

Universidade Federal da Paraíba  
Centro de Ciências e Tecnologia  
Coordenação de Pós-Graduação em Informática

Dissertação de Mestrado

**Sistemas Híbridos Neurosimbólicos,  
Estudo e Implementação**

Patrícia Santos Andrade

Campina Grande, Paraíba, Brasil

Universidade Federal da Paraíba  
Centro de Ciências e Tecnologia  
Coordenação de Pós-Graduação em Informática

Patrícia Santos Andrade

**Sistemas Híbridos Neurosimbólicos,  
Estudo e Implementação**

*Dissertação submetida ao curso de Pós-Graduação em Informática do Centro de Ciências e Tecnologia da Universidade Federal da Paraíba, como requisito parcial para a obtenção do grau de mestre em Informática.*

**Área de concentração:** Inteligência Artificial

**Orientador:** José Homero Feitosa Cavalcanti

Campina Grande, Março de 1997



A553s Andrade, Patricia Santos  
Sistemas híbridos neurosimbólicos, estudo e  
implementação / Patricia Santos Andrade. - Campina Grande,  
1997.  
87 f.

Dissertação (Mestrado em Informática) - Universidade  
Federal da Paraíba, Centro de Ciências e Tecnologia.

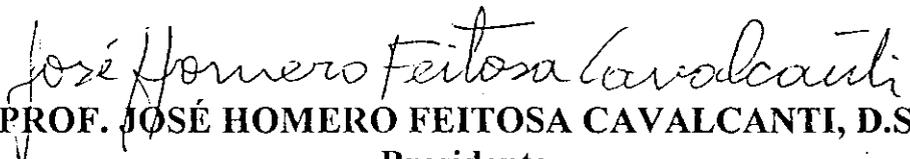
1. Inteligência Artificial - 2. Sistemas Híbridos  
Neurosimbólicos 3. Dissertação I. Cavalcanti, José Homero  
Feitosa, Dr. II. Universidade Federal da Paraíba - Campina  
Grande (PB)

CDU 004.8(043)

# Sistemas Híbridos Neuro-Simbólicos, Estudo e Implementação

Patrícia Santos Andrade

DISSERTAÇÃO APROVADA EM 12.03.1997

  
PROF. JOSÉ HOMERO FEITOSA CAVALCANTI, D.Sc  
Presidente

  
PROF. HÉLIO DE MENEZES SILVA, M.Sc  
Examinador

  
PROF. PABLO JAVIER ALSINA, D.Sc  
Examinador

  
PROF. EDSON COSTA DE BARROS CARVALHO FILHO, Ph.D  
Examinador

CAMPINA GRANDE - PB

Dedico este trabalho à meus pais,  
Maria e Odilon Andrade

## AGRADECIMENTOS

Gostaria de agradecer inicialmente a meus pais pela força e apoio que me deram e que foram indispensáveis para a realização desta dissertação.

Agradeço especialmente ao Prof. José Homero Feitosa Cavalcanti, Prof. Hermam Martins e Prof. Hélio Menezes pela orientação, conselhos, esclarecimentos e paciência. Agradecimentos também especiais, para as pessoas que moraram comigo durante algum tempo de minha permanência em Campina Grande: Milena, Gersa, Artur, José Eduardo, Julianelli, Danielli, Adriana, Jaque e Andreia, obrigado pelo companheirismo e amizade. Não poderia deixar de agradecer aos muitos amigos que fiz em Campina Grande: aos gaúchos, maranhenses, paraibanos, piauienses, alagoanos, baianos e cearenses. Agradeço também a todos os professores e funcionários da COPIN e as pessoas que contribuíram de alguma forma para a realização desta dissertação.

## RESUMO

As abordagens simbolista e conexionista são utilizadas para orientar o desenvolvimento de sistemas inteligentes. Cada abordagem possui vantagens e desvantagens. Muitas vezes as desvantagens de uma abordagem é compensada com a unificação com a outra abordagem. Este é o objetivo principal dos Sistemas Híbridos Neurosimbólicos. Este trabalho apresenta o SISNES, um sistema híbrido neurosimbólico implementado conforme a arquitetura de subprocessamento. Neste sistema as abordagens simbolista e conexionista são integradas de maneira que uma abordagem comanda os passos principais da resolução do problema e a outra está subordinada a este controle. A lógica Fuzzy também é utilizada neste sistema para o tratamento dos dados considerados incertos. O SISNES combina a grande capacidade de explicação dos Sistemas Especialistas Simbolistas, a robustez das Redes Neurais Artificiais, e a capacidade de representação ambígua da Lógica Fuzzy. Apresentam-se conceitos, arquiteturas, classificações e exemplos dos sistemas híbridos neurosimbólicos. Finalmente, apresentam-se resultados experimentais da implementação do SISNES na área de automação predial.

## ABSTRACT

The symbolist and connexionist approaches are utilized to guide the development of smart systems. Each approach has its advantages and its disadvantages. Many a time, the disadvantages of an approach are counterbalanced by the unification along with the advantages of the other approach. This is the main goal of the Neurosymbolic Hybrid Systems. This work presents SISNES as an implemented neurosymbolic hybrid system according to the subprocessing architecture. The symbolist and connexionist approaches in this system, are integrated so that an approach leads the main steps to the resolution of the problem while the other approach is subordinated to this very control. Fuzzy Logic is also used in this system, to processing data that are considered uncertain. SISNES combines great capability of explanation of the Symbolist Specialist Systems with the stoutness of the Neural Networks and with the ambiguous capability of representation of Fuzzy Logic. Concepts, architectures, classification and examples of the neurosymbolic hybrid systems are presented. Experimental results, from SISNES accomplishment on building automation scope, are finally presented.

## SUMÁRIO

### CAPÍTULO 1

INTRODUÇÃO .....	1
1.1 <i>Objetivos</i> .....	3
1.2 <i>Motivação</i> .....	3
1.3 <i>Relevância</i> .....	4
1.4 <i>Estrutura da Dissertação</i> .....	5

### CAPÍTULO 2

FERRAMENTAS PARA SISTEMAS INTELIGENTES .....	6
2.1 <i>A Abordagem Simbolista de Sistemas Inteligentes</i> .....	6
2.1.1 <i>Sistemas Especialistas</i> .....	7
2.2 <i>Abordagem Conexionista de Sistemas Inteligentes</i> .....	8
2.2.1 <i>Redes Neurais Artificiais</i> .....	9
2.2.2 <i>Neurônios Artificiais</i> .....	11
2.2.3 <i>Treinamento das Redes Neurais Artificiais</i> .....	13
2.3 <i>Lógica Fuzzy</i> .....	14
2.4 <i>Sistemas Híbridos Neurosimbólicos</i> .....	16

### CAPÍTULO 3

SISTEMAS HÍBRIDOS NEUROSIMBÓLICOS .....	19
3.1 <i>Classificações dos Sistemas Híbridos Neurosimbólicos</i> .....	19
3.2 <i>Arquiteturas dos Sistemas Híbridos Neurosimbólico</i> .....	22
3.2.1 <i>Arquitetura Híbrida Local/Distribuída</i> .....	22
3.2.2 <i>Arquitetura Pré/Pós Processamento</i> .....	24
3.2.3 <i>Arquitetura Subprocessamento</i> .....	25
3.2.4 <i>Arquitetura Metaprocessamento</i> .....	26
3.2.5 <i>Arquitetura Cooperação</i> .....	27

3.3 Exemplos de Sistemas Híbridos Neurosimbólicos .....	28
3.3.1 KBANN.....	28
3.3.2 Extensões do KBANN.....	31
3.3.3 INSS .....	33
3.3.4 SYNHESYS .....	34
3.4 Compilação e Extração de Regras.....	37
3.4.1 Compilação de Regras.....	38
3.4.2 Extração de Regras.....	41

## CAPÍTULO 4

O SISNES.....	45
4.1 A Arquitetura do SISNES.....	45
4.2 Classificação do SISNES.....	47
4.3 Os Módulos do SISNES.....	47
4.3.1 Aquisição e Base de Conhecimento.....	48
4.3.2 Motor de Inferência.....	49
4.3.3 Módulos de Explicação.....	49
4.3.4 Interface com o Usuário.....	50
4.4 Implementação da Integração Neurosimbólica.....	53
4.4.1 Código do LPA-Prolog.....	53
4.4.2 Código do C.....	54

## CAPÍTULO 5

APLICAÇÃO DO SISNES.....	55
5.1 Automação Predial.....	55
5.2 O Problema da Umidade.....	56
5.3 Codificação das Perguntas.....	59
5.4 Codificação das Regras.....	60
5.5 Inicialização do SISNES.....	63
5.6 Análise da Umidade do Edifício.....	65

---

5.6.1 Informações sobre os Andares do Edifício.....	66
5.6.2 Diagnóstico da Umidade no Edifício .....	67
5.6.3 Execução do Módulo Conexcionista.....	68
5.6.3.1 Explicação Conexcionista .....	74
5.6.4 - Utilização da Lógica fuzzy.....	76
<b>CAPÍTULO 6</b>	
CONCLUSÃO .....	80
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>83</b>

## LISTA DE FIGURAS

### CAPÍTULO 2

Fig. 2.1 - Arquitetura de um Sistema Especialista .....	8
Fig. 2.2 - Rede Neural Multi Camadas .....	10
Fig. 2.3 - Rede Neural Artificial de Hopfield .....	11
Fig. 2.4 - O Perceptron .....	11
Fig. 2.5 - O Neurônio de Hopfield.....	12
Fig. 2.6 - Função Linear .....	12
Fig. 2.7- Função Lógica.....	12
Fig. 2. 8 - Função Sigmoides .....	13
Fig. 2.9- Derivada da Sigmoides.....	13
Fig. 2.10- Função Fuzzy .....	16
Fig. 2.11 - Arquitetura Básica Híbrida .....	18

### CAPÍTULO 3

Fig. 3.1 - Classificação dos Sistemas Neurosimbólicos segundo Medsker e Bailey.....	20
Fig. 3.2 - Classificação dos Sistemas Neurosimbólicos segundo segundo Orsier .....	22
Fig. 3.3 - Arquitetura Local/Distribuída.....	23
Fig. 3.4 - Arquitetura Pré/Pós Processamento .....	24
Fig. 3.5 - Arquitetura Subprocessamento .....	25
Fig. 3.6 - Arquitetura Metaprocessamento .....	27
Fig. 3.7 - Arquitetura de Cooperação .....	28
Fig. 3.8 - Arquitetura do KBANN.....	30
Fig. 3.9 - Ciclo do KBANN.....	31
Fig. 3.10 - Arquitetura KBANN-DAID.....	32
Fig. 3.11 - Arquitetura do INSS .....	34
Fig. 3.12 - Arquitetura do SYNHESYS .....	35
Fig. 3.13 - Transformação das Regras para a hierarquia das Redes Neurais.....	40

Fig. 3.14 - Regras Extraídas da Ativação de um Neurônio .....	41
Fig. 3.15 - Neurônio com seus Pesos e Conexões .....	42
Fig. 3.16 - Neurônio com seus Pesos e Conexões .....	43
Fig. 3.17 - Neurônio com seus Pesos e Conexões .....	43

#### CAPÍTULO 4

Fig. 4.1 - Arquitetura de Subprocessamento do SISNES .....	46
Fig. 4.2- Comunicação entre o Módulo Simbólico e o Conexionista .....	46
Fig. 4.3 - Localização do SISNES no diagrama de Orsier .....	47
Fig. 4.4 - Módulos da Arquitetura do SISNES.....	48
Fig. 4.5 - Aquisição do Conhecimento no SISNES.....	49
Fig. 4.6 - Comportamento do SISNES para o Diagnóstico .....	51
Fig. 4.7 - Comportamento do SISNES para a execução da Rede Neural.....	52

#### CAPÍTULO 5

Fig. 5.1 - Tela Inicial do SISNES.....	63
Fig. 5.2 - Tela de Aplicações do SISNES.....	64
Fig. 5.3 - Tela de Manutenção do SISNES.....	65
Fig 5.4 - Tela do Diagnóstico de Umidade em Edificações .....	66
Fig. 5.5 - Tela das Perguntas para o Diagnóstico no Subsolo .....	66
Fig. 5.6 - Tela do Resultado Final do SISNES.....	67
Fig. 5.7 - Tela da Explicação Simbólica do SISNES .....	68
Fig. 5.8 - Tela para Execução do Módulo Conexionista .....	69
Fig. 5.9 - Tela para Seleção da Entrada de Rede Neural .....	69
Fig. 5.10 - Rede Neural do Módulo Conexionista.....	70
Fig. 5.11 - Conjunto de Treinamento da Rede Neural.....	71
Fig. 5.12 - Resultado da Rede Neural.....	72
Fig. 5.13 - Tela do Resultado Conexionista .....	73
Fig. 5.14 - Inversa da Rede Neural .....	74
Fig. 5.15 - Conjunto de Treinamento da Falsa Inversa .....	75

Fig. 5.16 - Explicação do Resultado da Rede Neural.....	75
Fig. 5.17 - Tela da Explicação Conexionista.....	76
Fig. 5.18 - Tela de Solicitação da Entrada Fuzzy .....	77
Fig. 5.19 - Função de Pertinência Fuzzy .....	78
Fig. 5.20 - Tela do Resultado Fuzzy.....	79

## LISTA DE TABELAS

### **CAPÍTULO 2**

Tab. 2.1 - Desvantagens do Simbolismo e Vantagens do Conexionismo ..... 17

Tab. 2.2 - Desvantagens do Conexionismo e Vantagens do Simbolismo ..... 17

### **CAPÍTULO 3**

Tab. 3. 1 - Classificação dos Sistemas Neurosimbólicos conforme o Meio de Comunicação20

Tab. 3. 2 - Tabela de Mapeamento Regras-Redes do KBANN..... 29

### **CAPÍTULO 5**

Tab. 5.1 - Resultados da Rede Neural do Módulo Conexionista ..... 71

## SIMBOLOGIA

### Caracteres latinos e gregos:

- Wi - Pesos das conexões entre os neurônios
- $\Sigma$  - Somatório utilizado nas redes neurais
- $\alpha$  - Declividade da função linear
- $\theta$ ,  $\beta$  e BIAS - Parâmetros dos Neurônios
- Tf - Função de pertinência fuzzy da variável T
- Sn - Codificação das perguntas do SISNES
- Ri - Codificação das regras no SISNES
- SAI - Saída da rede neural
- X - Entrada da rede neural
- MCTBP, MCBTP, NHMP, MUTP - Possíveis diagnósticos do SISNES
- TS - Teor de Sais na parede
- TSf - Teor de Sais obtido pela lógica fuzzy
- A - Declividade das funções de pertinência fuzzy
- Bp e Bn - Deslocamento das funções de pertinência fuzzy
- Z,P,M e G - Valores das funções de pertinência fuzzy

### Abreviações:

- RNMC - Rede Neural Multicamadas
- RNH - Rede Neural de Hopfield
- RBC - Raciocínio Baseado em Casos
- SYNHESYS, KBANN, INSS e SISNES- Sistemas híbridos neurosimbólicos
- DAID - Extensão do KBANN
- QUICKPROP - Algoritmo de treinamento das redes neurais artificiais
- MC - Módulo Conexionista
- MS - Módulo Simbolista
- SC - Solução do módulo Conexionista

SS - Solução do módulo Simbolista

SE - Solução do Especialista

SUBSET e NofM - Algoritmos para extração de regras das redes neurais artificiais

# Capítulo 1

## INTRODUÇÃO

A Inteligência Artificial trata a resolução de problemas como um conjunto de técnicas e definições com a finalidade de orientar o desenvolvimento de sistemas que simulam atividades humanas, usualmente consideradas inteligentes. Estes sistemas são chamados de Sistemas Inteligentes [RIC94].

Nos sistemas inteligentes, o conhecimento é adquirido na fase de aprendizagem, e utilizam técnicas para tratamento das informações (por exemplo: estatísticas, probabilidades, lógica fuzzy, etc.). O processo de aprendizado segue uma metodologia. Uma opção é descrever como identificar um objeto fornecendo informações dos componentes, formas, características importantes e relacionamentos entre as informações fornecidas. No final do aprendizado, existirão condições de reconhecer um objeto, através de interações entre as informações dadas e o que foi aprendido. Esta forma de aprendizado está baseada no aprendizado teórico<sup>1</sup> (sem exemplos). Outra opção é mostrar ao aprendiz vários objetos que pertençam a certa classe até que ele tenha condições de reconhecer um novo objeto. Esta outra forma de aprendizado é baseada em exemplos (sem definições teóricas) [TOW91A]. Recentemente a ênfase dos

---

<sup>1</sup> Aprendizado que utiliza o conhecimento teórico representado através de regras por exemplo: regras e definições do objeto.

pesquisadores concentra-se nas abordagens de aprendizagem híbrida que utilizam o aprendizado teórico e um conjunto significativo de exemplos acerca do domínio do problema [TOW 94]. O exemplo mais amplo de sistemas inteligentes são os sistemas especialistas.

Os sistemas especialistas têm por finalidade solucionar problemas complexos do mundo real com o comportamento similar ao de um especialista humano. Para solucionar problemas, os sistemas especialistas precisam ter acesso a uma base de conhecimento de forma eficiente e correta, precisam de um mecanismo de raciocínio que oriente o processo de resolução do problema e de um mecanismo que explique o seu resultado final. Os sistemas especialistas são desenvolvidos pelo engenheiro do conhecimento, que o planeja e o desenvolve, colocando nele todo o conhecimento que foi extraído de um especialista humano. O sistema especialista mais conhecido é o MYCIN, que é baseado em regras. Ele usa o raciocínio inexato (estatística), e é aplicado na diagnose médica de terapias apropriadas para pacientes com infecções bacterianas.[RIC94]

As principais abordagens de sistemas inteligentes são a simbolista e a conexionista. Os sistemas desenvolvidos conforme o simbolismo são baseados no aprendizado teórico, dependem da aquisição de conhecimento. Eles são mais adequados para problemas de dedução lógica e possuem boa capacidade de explicação. Um exemplo de um sistema inteligente simbolista é o MYCIN. Os sistemas desenvolvidos conforme o conexionismo são baseados no aprendizado através de exemplos. Eles são mais adequados para tarefas de reconhecimento de padrões e neles as decisões levam em consideração a experiência acumulada. Os sistemas conexionistas usam as redes neurais artificiais que simulam o funcionamento do cérebro humano[DON94]. A idéia básica das redes neurais artificiais é criar uma máquina inteligente com neurônios artificiais interconectados.

A lógica “fuzzy” (lógica nebulosa) é aplicada, para o tratamento das informações, nas abordagens simbolista e conexionista. Isso se deve principalmente à sua capacidade de representar e tratar as incertezas, normalizar e padronizar parâmetros. [KLE 88][ZAD93][ZIM 91].

Alguns trabalhos mostram os problemas que inviabilizam a classificação de qualquer uma das abordagens apresentadas como uma metodologia de aprendizado de propósito geral [KAS 90] [TOW 91A] [GIA 92] [ORS 95]. A utilização de uma abordagem híbrida que combine as características simbolistas e conexionistas criando condições para que trabalhem cooperativamente, tirando proveito das capacidades e vantagens de cada abordagem e

minimizando as suas deficiências e desvantagens, surge como uma nova abordagem, denominada de sistemas híbridos neurosimbólicos. Esta abordagem híbrida pretende ser de qualidade superior, com utilização mais geral que as abordagens isoladas e pretende favorecer o desenvolvimento de sistemas especialistas mais robustos e eficientes [LOP93][OSO95].

Este trabalho apresenta o projeto, implementação e análise de um sistema híbrido neurosimbólico chamado SISNES, que utiliza regras, redes neurais artificiais e lógica fuzzy. Apresenta-se também a aplicação experimental do SISNES na área de edificações, mais especificamente no diagnóstico da umidade na estrutura de alvenaria de edifícios de múltiplos andares.

## 1.1 Objetivos

Os objetivos desta dissertação são os seguintes:

1. Realizar pesquisa e estudo sobre sistemas híbridos neurosimbólicos incluindo os principais conceitos, definições, classificações, arquiteturas e exemplos de sistemas já desenvolvidos.
2. Estudar as características dos sistemas híbridos neurosimbólicos já existentes, para cada um, identificar a sua aplicação, classificação e arquitetura.
3. Propor e implementar um sistema híbrido neurosimbólico para uma aplicação específica.
4. Classificar o novo sistema híbrido neurosimbólico conforme os conceitos e arquitetura pesquisados e estudados.

## 1.2 Motivação

Nas últimas décadas, foram desenvolvidas muitas pesquisas na área de aprendizagem de máquina (sistemas inteligentes), proporcionando o desenvolvimento das abordagens simbolista e conexionista de uma forma independente. Recentemente as pesquisas estão se tornando menos independentes e cada vez mais cresce a quantidade de pesquisas sobre a integração das

duas abordagens [SHA 92]. Considera-se que três artigos publicados no IJCAI-89<sup>2</sup> ([Fis 89] [Moo 89] [Wei 89]) começaram os estudos e a comparação entre as abordagens simbolista e conexionista, e que logo foram seguidos por outros pesquisadores. Atualmente a integração híbrida neurosimbólica é o foco principal do projeto MIX<sup>3</sup> [HL 94] na Suíça e na França, que tem como propósito investigar as estratégias e desenvolver ferramentas para a integração dos componentes simbólicos e conexionistas.

Através de discussões em *Workshops*, Congressos e Seminários sobre sistemas híbridos neurosimbólicos, criou-se um consenso entre os pesquisadores de que a nova abordagem híbrida constitui um promissor caminho no desenvolvimento de sistemas inteligentes mais robustos, poderosos e com maior versatilidade. As motivações básicas para a continuação de pesquisas na abordagem híbrida são [SUN94]:

1. O processo de aprendizagem não é homogêneo e uma grande variedade de representações e mecanismos estão envolvidos. Alguns aspectos do processo de aprendizagem dos sistemas inteligentes são melhor representados na abordagem simbolista, enquanto outros aspectos na abordagem conexionista. Então, existe a necessidade de um pluralismo entre os processos de aprendizagem que orientam o desenvolvimento dos sistemas inteligentes no sentido de fornecer ferramentas e técnicas híbridas.
2. O desenvolvimento de sistemas híbridos neurosimbólicos para diferentes aplicações práticas pode propiciar o surgimento de novas técnicas para integrar as abordagens simbolista e conexionista.

### 1.3 Relevância

Esta dissertação tem como expectativa principal de contribuição:

1. O estudo da nova abordagem de construção de sistemas inteligentes, os sistemas híbridos neurosimbólicos.
2. Implementar uma aplicação específica

---

<sup>2</sup> International Joint Conference on Artificial Intelligence.

<sup>3</sup> Modular Integration of Connectionist and Symbolic Processing in Knowledge-Based Systems.

Esta dissertação servirá como referência a outras dissertações ou a trabalhos em geral com temas similares, e fornecerá suporte a futuros trabalhos em relação ao levantamento bibliográfico e aos estudos dos sistemas já existentes. Além disso espera-se que ela contribua no embasamento teórico no assunto, visto que se refere a uma nova abordagem com pesquisas ainda em andamento.

## 1.4 Estrutura da Dissertação

Esta dissertação é dividida em 5 (cinco) capítulos que tratam sobre um estudo e uma implementação de sistemas híbridos neurosimbólicos.

No capítulo 1 é feita uma introdução ao assunto que este trabalho trata e apresentado seu objetivo, motivação e relevância.

No capítulo 2 são apresentados alguns conceitos da abordagem simbolista e da abordagem conexionista de sistemas inteligentes. Além de definições de Sistemas Especialistas, Redes Neurais Artificiais e Lógica Fuzzy e uma introdução aos Sistemas Híbridos Neurosimbólicos.

No capítulo 3 são apresentadas classificações e arquiteturas dos sistemas híbridos neurosimbólicos e exemplos dos principais sistemas implementados, suas arquiteturas, classificações e aplicações.

No capítulo 4 é apresentado o sistema híbrido neurosimbólico proposto, denominado SISNES, sua arquitetura e classificação.

No capítulo 5 é apresentado os resultados da implementação do SISNES na área de automação predial.

No capítulo 6 são apresentadas as conclusões desta dissertação de mestrado e sugestões para trabalhos a serem desenvolvidos.

## Capítulo 2

### FERRAMENTAS PARA SISTEMAS INTELIGENTES

Este capítulo apresenta aspectos das abordagens simbolista e conexionista no desenvolvimento de sistemas inteligentes. A seguir serão apresentados sucintamente alguns conceitos referentes ao desenvolvimento de sistemas inteligentes. Os conceitos deste capítulo têm por finalidade apresentar noções básicas para os assuntos: sistemas especialistas, redes neurais artificiais, lógica “fuzzy” e sistemas híbridos neurosimbólicos.

#### 2.1 A Abordagem Simbolista de Sistemas Inteligentes

Os sistemas inteligentes que são desenvolvidos com base no aprendizado teórico fazem parte da abordagem simbolista. Na fase de aprendizado utilizam quase exclusivamente o domínio teórico<sup>4</sup> [TOW91A][TOW91B]. As principais desvantagens desta abordagem são:

1. *Pressupõe-se que o domínio teórico esteja completo e correto* [RIC94]. Os algoritmos básicos dos sistemas inteligentes desenvolvidos baseados no aprendizado teórico assumem que o domínio teórico esteja completo e correto. Desta forma, tudo o que se pode deduzir com base no domínio teórico está correto e o que não se pode deduzir

---

<sup>4</sup> O domínio teórico geralmente é uma coleção de regras que descrevem as interações dos fatos do problema

não faz parte da questão em consideração. Construir um domínio teórico correto e completo é extremamente difícil e vagaroso, em alguns casos é impossível [TOW 91A], como por exemplo, modelar os movimentos de um robô onde as condições circunvizinhas estão constantemente mudando.

2. *O domínio teórico pode ser computacionalmente intratável* [TOW 91A]. Sistemas baseados no aprendizado teórico geralmente processam suas soluções utilizando a construção de provas que necessitam de muito tempo e memória computacionais [RIC 94]. Um sistema deste tipo, apesar de possuir o domínio teórico correto, pode estar incapacitado no momento de processar as soluções por um impedimento computacional.
3. *A forma de representação do domínio teórico é um fator limitante* [RIC 94]. A criação do domínio teórico adequado para ser usado nos sistemas simbolistas é um significativo problema de aprendizagem e um fator determinante no desenvolvimento de sistemas inteligentes.

### 2.1.1 Sistemas Especialistas

Os Sistemas Especialistas são geralmente desenvolvidos conforme a abordagem simbolista. Nos sistemas especialistas o domínio teórico é chamado de base de conhecimento e é adquirido através do aprendizado teórico. A base de conhecimento é geralmente organizada como uma combinação de estruturas de dados, procedimentos de interpretação e inferências. As representações da base de conhecimento mais usadas são: redes semânticas<sup>5</sup>, regras de produção e *frames*<sup>6</sup>. [HAR 90] [LOP 93]. A Fig. 2.1 mostra a arquitetura básica de um sistema especialista.

A aquisição do conhecimento é uma etapa do processo de desenvolvimento do sistema especialista. Ela é responsável por extrair o domínio teórico do especialista humano e formar a

---

<sup>5</sup> Forma de representação do conhecimento simbólicos na forma hierárquica de árvores, é basicamente um grafo. São compostas por nós representando as entidades ou objetos do domínio teórico, e elos representando as relações entre os nós. Possui a propriedade de herança dos nós que estão hierarquicamente acima.

<sup>6</sup> São uma extensão das redes semânticas. Permitem que os nós possuam estruturas internas. Os nós são compostos de um nome, uma lista de atributos e seus valores.

base de conhecimento[HAR 90]. A arquitetura de um sistema especialista (Fig. 2.1) é composta por módulos com funções bem definidas. Uma solução é produzida com base no domínio teórico localizado na base de conhecimento. O módulo chamado motor de inferência é formado por um algoritmo que controla e decide qual será o próximo passo para a determinação da solução. O módulo de explicação é responsável, com o auxílio das informações do motor de inferência, pela análise e seleção da seqüência de regras e fatos que melhor expliquem a solução.

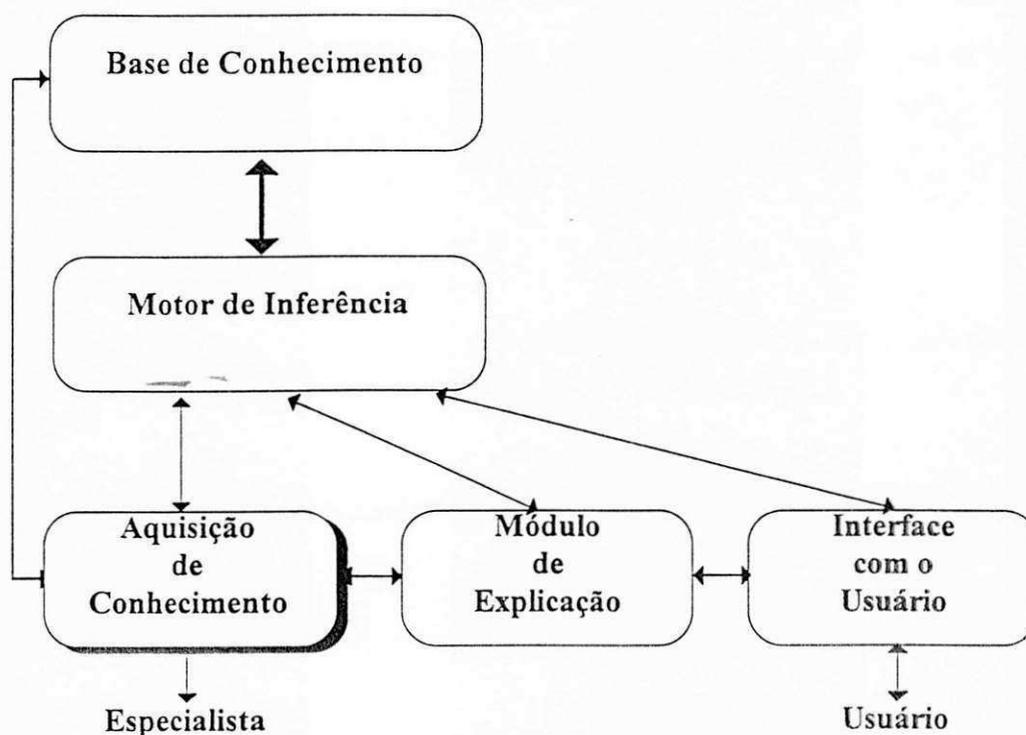


Fig. 2.1 - Arquitetura de um Sistema Especialista

## 2.2 Abordagem Conexionista de Sistemas Inteligentes

Considera-se que os sistemas inteligentes que são desenvolvidos com base em exemplos geralmente fazem parte da abordagem conexionista. Na fase de aprendizado eles utilizam um conjunto de exemplos geralmente grande e possivelmente incompleto, e quase nenhuma teoria[TOW91A]. As principais desvantagens desta abordagem são:

1. *Induções Incorretas.* Características parecidas podem conduzir a uma classificação incorreta [BEA 90]. Por exemplo, se na fase de aprendizado é apresentado ao

aprendiz. objetos de uma só cor; e se for apresentado o mesmo objeto com uma cor diferente, ele poderá não ser classificado corretamente [TOW91A].

2. *Representação das exceções.* Quando um conjunto muito grande de exemplos é utilizado, um pequeno conjunto de exceções pode não ser representado ou ser representado de uma maneira inadequada [BEA90] [TOW91A].
3. *Características importantes dependem do contexto.* Um grande número de características pode descrever um objeto. Dois objetos distintos podem se tornar similares ou não, dependendo das características que serão escolhidas para descrevê-los. Desta forma, pode-se notar que, o aprendizado através de exemplos precisa de informações que indiquem as características relevantes ao contexto do problema [TOW91A].
4. *Características irrelevantes.* Na descrição dos exemplos as características irrelevantes podem ter efeitos negativos [TOW91A]. Por exemplo, se a cor de um objeto for uma característica irrelevante para determinado problema, e na descrição dos exemplos for considerada relevante, o número de possíveis classificações incorretas poderá crescer.

### 2.2.1 Redes Neurais Artificiais

As redes neurais artificiais possuem processamento distribuído e paralelo. O primeiro modelo matemático de redes neurais artificiais foi proposto por McCulloch e Pitts em 1943 [RIC94]. A rede neural artificial do tipo multicamadas ou redes de Rumelhart [RUM89], é o exemplo mais comum usado na abordagem conexionista. A Fig.2.2 mostra a arquitetura das Redes Neurais artificiais do tipo Multi Camadas (RNMC) que são compostas por uma coleção de neurônios artificiais conectados em camadas e com o fluxo de dados num único sentido. A RNMC da Fig.2.2 possui três camadas: camada de entrada, camada intermediária (ou escondida) e camada de saída.

Outro tipo de rede neural artificial é a Rede Neural artificial de Hopfield (RNH) [BEA90] [RUM89] [HOP82][HOP84], mostrada na Fig.2.3, nesta rede os neurônios são interconectados, eles não são dispostos em camadas e o fluxo dos dados na rede é multi-direcional.

Hopfield definiu uma rede neural artificial clara, determinística e que podia ser comparada com o cérebro humano. A equação do modelo individual dos neurônios é simples, mas quando eles são conectados podem se tornar sistemas complexos. A conexão entre os neurônios é simétrica tal que os pesos  $W_{ij}=W_{ji}$ , e somente um neurônio pode mudar de estado em um instante de tempo. A escolha do neurônio mutante é aleatória. Hopfield demonstrou que existe uma função computacional para o seu modelo que diminui com a evolução do tempo, levando os neurônios da rede a um estado estável.

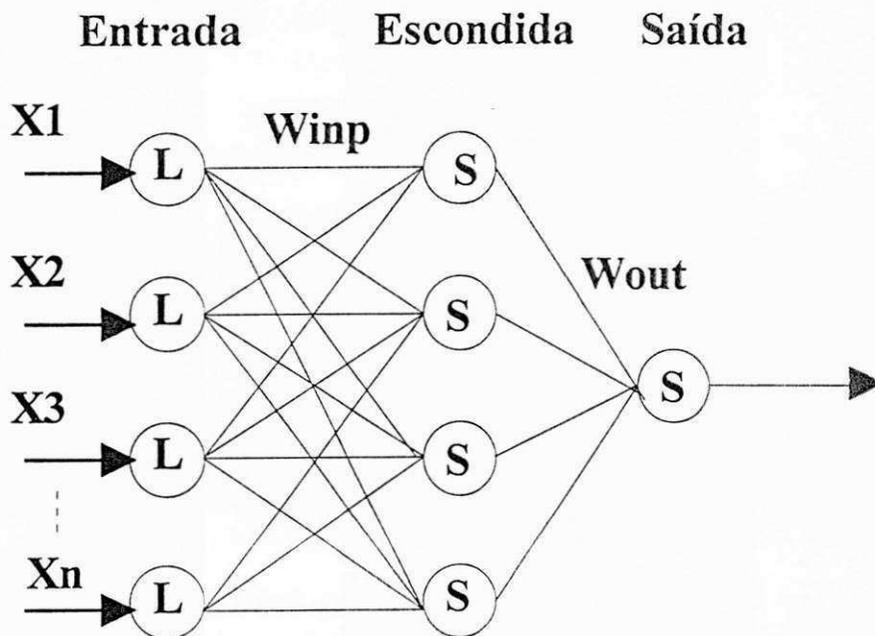


Fig. 2.2 - Rede Neural Multi Camadas

A RNH mostrada na Fig.2.3 possui três neurônios com entradas excitatórias comuns ( $I=2k-1$ ,  $k$  o número de neurônios vencedores), ela está projetada na forma K-WINNER, em que  $k$  neurônios ficarão no estado "1" e  $3-k$  neurônios no estado "0".

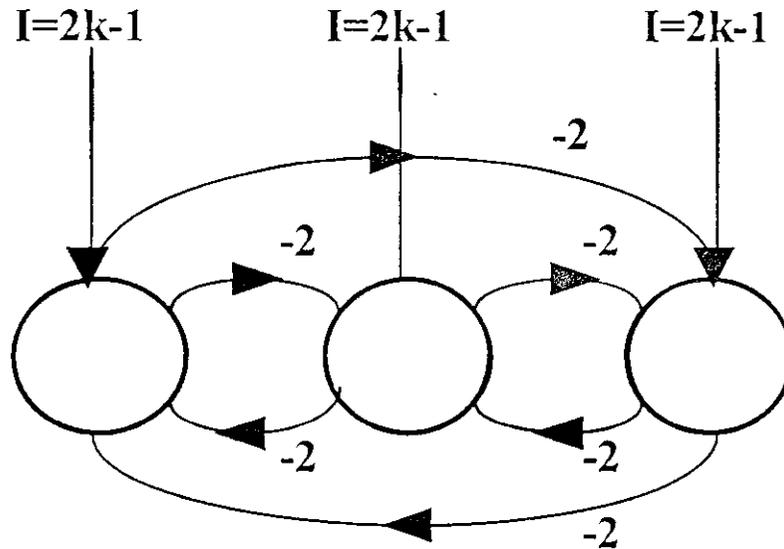


Fig. 2.3 - Rede Neural Artificial de Hopfield

### 2.2.2 Neurônios Artificiais

Os neurônios artificiais são unidades de processamento local que dependem somente dos valores nas suas entradas. Os neurônios implementam uma função e estabelecem parâmetros para o cálculo do valor de saída da função. Na Fig.2.4 é mostrado o neurônio artificial PERCEPTRON, proposto por Rosenblat [ROS57], que é capaz de aprender de acordo com as regras propostas por Hebb. As entradas dos neurônios são representadas por  $X_i$ , os pesos das entradas por  $W_i$ , e a saída intermediária  $Y(t)=\sum X_i W_i$ . O valor de saída do neurônio é o seguinte:  $Z(t)=f(Y(t))$ . Geralmente a função  $f(\cdot)$  é não linear.

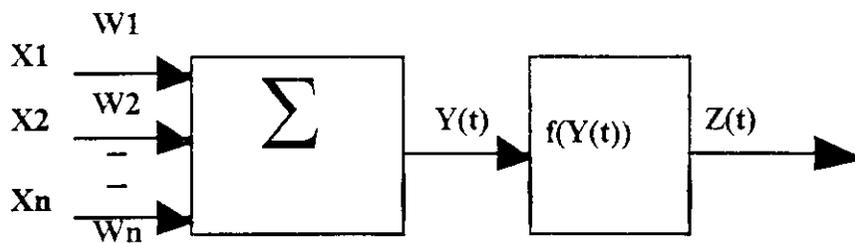


Fig. 2.4 - O Perceptron

A Fig. 2.5 mostra o esquema do neurônio de Hopfield com suas entradas, saída, função e parâmetros. Ele é semelhante ao Perceptron. Ele se caracteriza por um integrador na sua saída, o que permite a memorização dos estados anteriores.

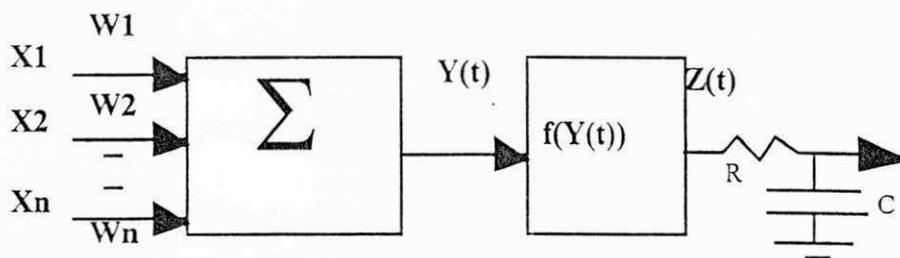


Fig. 2.5 - O Neurônio de Hopfield

A depender do tipo de função que o neurônio implementa ele pode ser classificado, entre outros tipos, em: linear, lógico e sigmoide. O neurônio linear implementa uma função linear do tipo:  $Z=f(Y)=\alpha \sum X_i W_i$ , a Fig.2.6 mostra o comportamento da função linear implementada por este tipo de neurônio.

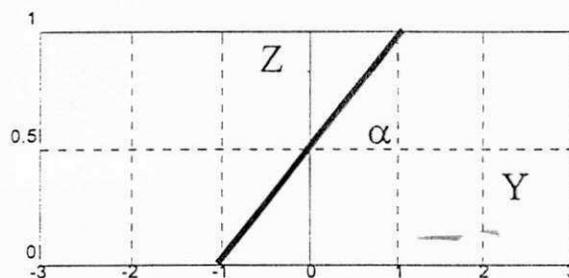


Fig. 2.6 - Função Linear

O neurônio lógico, também chamado de neurônio binário, implementa uma função do tipo:  $Y(t)=\sum X_i W_i$  e uma regra de controle do tipo:

if  $Y(t) > 0$  then  $Z(t)=1$ ;  
else  $Z(t)=0$ ;

A Fig. 2.7 mostra o comportamento da função implementada por este tipo de neurônio.

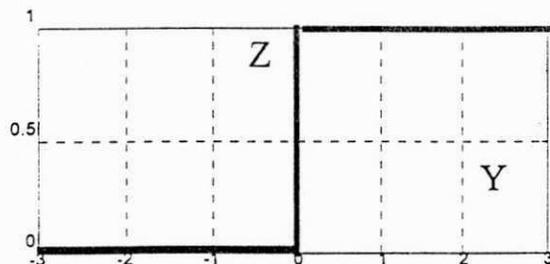
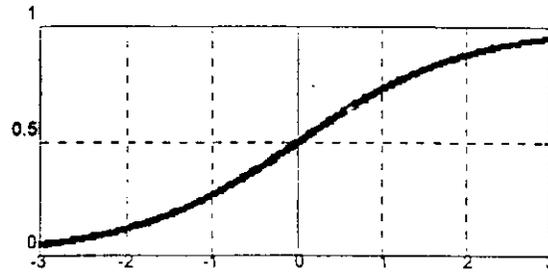
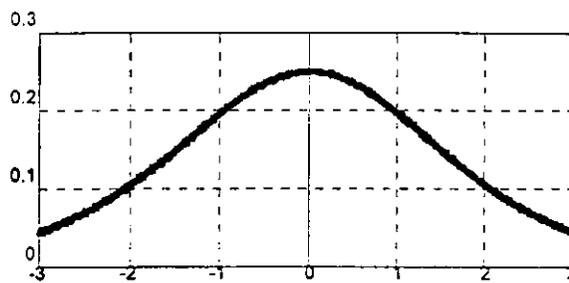


Fig. 2.7- Função Lógica

O neurônio do tipo sigmoide implementa uma função sigmoide:  $Z = \theta / [1 + \exp(-\beta Y)]$ , onde  $Y = \sum X_i W_i + T$ . A Fig.2.8 mostra o comportamento da função sigmoide ( $\theta = 1$ ,  $\beta = 1$  e  $T = 0$ ). Na Fig.2.9 é mostrada a curva da derivada da função sigmoide.



**Fig. 2. 8 - Função Sigmoide**



**Fig. 2.9- Derivada da Sigmoide**

### 2.2.3 Treinamento das Redes Neurais Artificiais

A cada neurônio estão associadas conexões com outros neurônios de outras camadas ou com neurônios da mesma camada. Cada conexão possui um peso e são chamadas de *sinapses*. Cada *sinapse* pode ser classificada como excitatória, quando atua sobre o valor de entrada positivamente, ou inibitória, quando atua sobre o valor de entrada negativamente [BEA 90].

O conhecimento nas redes neurais é representado pelas conexões e parâmetros dos neurônios, o que torna difícil determinar uma explicação para as soluções produzidas. Geralmente as redes neurais adquirem conhecimento através do aprendizado por exemplos, tendo como base um conjunto significativo de exemplos acerca do domínio do problema. Este conjunto de exemplos pode ser adquirido automaticamente a partir do problema ou fornecido por um especialista [BEA 90].

Para treinar uma rede neural é necessário ajustar os pesos das conexões, aumentando ou diminuindo seus valores de acordo com um índice de desempenho predefinido. Para realizar o

aprendizado em uma rede não treinada ou para refinar o seu conhecimento é definido um algoritmo de treinamento, que ajusta os pesos das *sinapses* com o objetivo de diminuir o erro da rede neural [RUM89]. O algoritmo de treinamento mais utilizado em redes neurais multicamadas é o de retropropagação.

Alguns trabalhos [RUM89][SCH93][SCH95] mostram que o algoritmo de retropropagação é simples e que pode sofrer uma série de melhorias. A aplicação do algoritmo de aprendizagem de retropropagação, restringe algumas possibilidades interessantes das redes neurais artificiais e possui algumas propriedades importantes que dificultam o aprendizado. Os principais problemas do uso deste algoritmo são:

1. Dificuldade na especificação da topologia da rede;
2. Dificuldade para a determinação dos parâmetros que controlam a aprendizagem;
3. Problemas de convergência;
4. Aprendizado não incremental (seqüencial) que obriga a apresentar toda a base de exemplos a cada passo, se o algoritmo não for on-line;
5. Problemas de competição entre unidades.

## 2.3 Lógica Fuzzy

O diálogo e o raciocínio humano são nebulosos (“fuzzy”). Características como incerteza, incompletude, inconsistência e imprecisão estão presentes na vida diária dos homens. A teoria dos conjuntos nebulosos formalizou a comunicação nebulosa através da lógica fuzzy, tornando a comunicação homem - máquina mais eficiente [KLE88][ZIM91].

A lógica fuzzy difere da lógica clássica porque tenta modelar a imprecisão do raciocínio que está presente na capacidade humana de inferir sobre respostas aproximadas. Ela tenta aplicar este tipo de raciocínio nas questões que são baseadas num conhecimento que não é exato, que está incompleto ou não está totalmente disponível e tomar decisões racionais neste ambiente de incertezas [ZAD 93]. O tratamento dos dados e parâmetros das abordagens simbolistas e conexionistas podem passar por um processo de fuzziificação<sup>7</sup> ou desfuzziificação<sup>8</sup> para melhor representar o raciocínio humano.

---

<sup>7</sup> Transformação de um valor numérico para um valor lingüístico

<sup>8</sup> Transformação de um valor lingüístico para um valor numérico

As principais características da lógica fuzzy são [ZAD93]:

1. Os valores verdade são associados a vários subconjuntos fuzzy. Por exemplo, se  $T$  é um intervalo numérico ( $0 \leq T \leq 1$ ), então o valor verdade correspondente a *grande-G* ( $\geq 0,8$ ) pode ser interpretado como um subconjunto deste intervalo. Neste sentido, um valor verdade em fuzzy pode ser visto como uma representação de vários valores numéricos.
2. Os predicados podem ser limitados para verdadeiro ou falso, como por exemplo, *mortal*, *sempre*, e *pai de*, ou podem ser mais gerais como por exemplo: *doente*, *cansado*, *alto*, *largo*, *amigo de*, etc..
3. Ela permite usar além dos quantificadores *algum* e *todo* da lógica clássica quantificadores adicionais como: *o mais*, *muitos*, *vários*, *poucos*, *alguns de*, *frequentemente*, *ocasionalmente*, *por volta de dez*, etc.
4. Ela permite trabalhar com variáveis lingüísticas, isto é, variáveis cujo valores são palavras ou sentenças em linguagem natural. Por exemplo, *idade* é uma variável lingüística e os valores que são assumidos podem ser *jovem*, *velho*, *não tão velho*, etc.

Uma proposição na lógica fuzzy é formada da seguinte forma:  $P \rightarrow X \text{ é } A$ , onde  $X$  é uma variável e  $A$  é um predicado fuzzy. Esta proposição  $P$  implica a possibilidade de  $X$  se relacionar com o predicado fuzzy  $A$ , que é um subconjunto fuzzy e onde  $f_a(u)$  é a função de relacionamento do predicado fuzzy  $A$  com o valor de entrada igual ao valor numérico  $u$ . Por exemplo se  $P \rightarrow \text{João é alto}$  então  $X = \text{Altura}(\text{João})$ ,  $A = \text{Alto}$  (subconjunto fuzzy),  $f_{\text{alto}}$  é a função de relacionamento de alto e  $f_{\text{alto}}(u)$  é o grau do relacionamento[ZAD93].

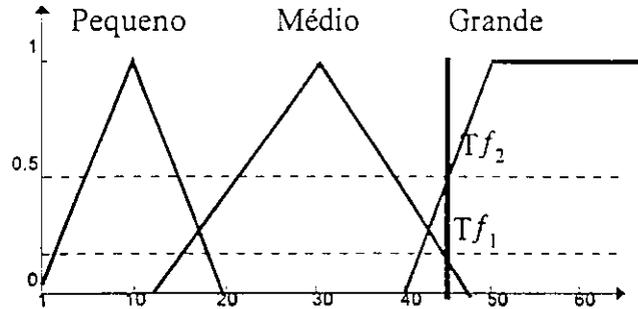
Um outro exemplo seria a regra abaixo, extraída de um sistema especialista que realiza o diagnóstico da umidade em edificações, desenvolvido segundo a abordagem simbolista[SOU 94]. Nesta regra  $T = \text{TEOR\_DE\_SAIS\_NA\_PAREDE}$  e  $R = \text{resposta}$ , seria:

**SE T = 45 ENTÃO R = HIGROSCOPICIDADE**

Esta regra poderia sofrer uma transformação (fuzzificação). Na Fig. 2.10 vê-se que para  $T = 45$ , os valores fuzzy das suas funções de pertinência são  $Tf_1 = 0,2$  e  $Tf_2 = 0,5$ . O teor 45 se aproxima dos conjuntos dos teores definidos como teores médios ( $Tf_1$ ) e grandes ( $Tf_2$ ). O vencedor é  $Tf_2$ , ( $Tf_2 > Tf_1$ ), o teor 45 é Grande e a nova regra seria:

**SE  $Tf$  = Grande ENTÃO R=HIGROSCOPICIDADE**

Onde Grande representa o intervalo de teores que podem ser considerado grandes.



**Fig. 2.10- Função Fuzzy**

Após a fuzzificação, a regra fuzzy se aproxima do vocabulário usual humano. Conhecendo-se os valores lógicos das variáveis fuzzy, a lógica fuzzy pode ser usada para a execução das regras.

Na abordagem conexionista, a lógica fuzzy pode ser usada para a normalização dos dados de entrada e de saída da rede neural. Neste trabalho a lógica fuzzy foi utilizada no auxílio do tratamento dos dados nas abordagens conexionista e simbólica

## 2.4 Sistemas Híbridos Neurosimbólicos

Um sistema híbrido representa uma integração em um mesmo ambiente de técnicas ou abordagens distintas. Por exemplo, os sistemas híbridos neurosimbólicos que combinam algoritmos genéticos, lógica fuzzy, redes neurais artificiais e sistemas especialistas [GOO 95]. Esta unificação tem como objetivo principal aproveitar as melhores propriedades e compensar as limitações de cada abordagem [GOO 95]. A integração pode ser feita de várias formas, por exemplo em 1988, Utgoff desenvolveu um algoritmo que integrou <sup>9</sup>árvores de decisão [RIC 94] e <sup>10</sup>perceptrons [RUM89] [UTG 88]. Outra maneira de integração é implementar no nível mais alto

<sup>9</sup> Um tipo de estrutura utilizada para representar o conhecimento teórico no método simbolista.

<sup>10</sup> Unidades de representação de neurônios do método conexionista.

da tomada de decisão a abordagem simbolista e no nível mais baixo a abordagem conexionista [GAL88].

Alguns trabalhos [TOW 91A] [GIA 92] [ORS 95A] mostram a unificação das abordagens simbolistas e conexionistas e obtiveram como resultado um sistema híbrido neurosimbólico com melhor desempenho quando comparado com outros sistemas desenvolvidos utilizando uma das abordagens isoladas. Uma característica importante, em particular das abordagens simbolista e conexionista, é que em alguns aspectos eles se complementam, a desvantagem de um é uma vantagem do outro, e vice-versa, como mostram as tabelas 2.1 e 2.2 [LOP93] [OSO95].

Desvantagens do Simbolismo	Vantagens do Conexionismo
Dificuldade na construção da base de conhecimento.	Aprendizado à partir de exemplos, fácil aquisição de conhecimento.
Fragilidade frente a informações aproximadas e incompletas.	Capacidade de generalização.
Processamento usualmente seqüencial e lento.	Possibilidade de paralelismo.

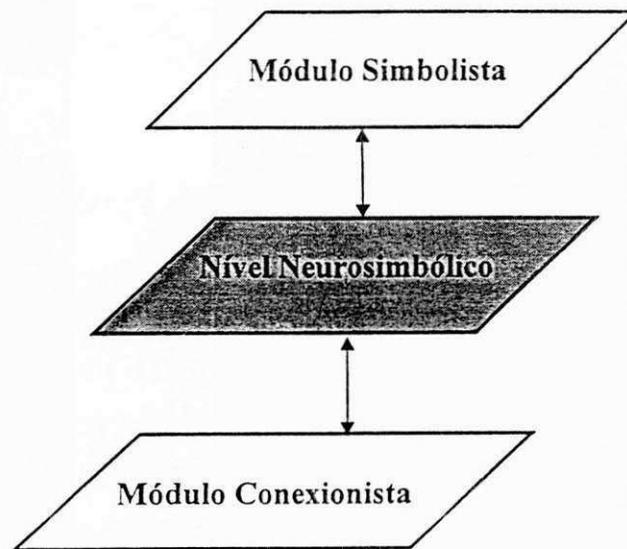
**Tab. 2.1 - Desvantagens do Simbolismo e Vantagens do Conexionismo**

Desvantagens do Conexionismo	Vantagens do Simbolismo
Dificuldade de determinação dos parâmetros de aprendizagem e da topologia da rede em função do problema. Aprendizado depende dos estados iniciais da rede.	Permite a inserção direta dos conhecimentos a priori sobre o problema.
O conhecimento adquirido é codificado em pesos e conexões de difícil interpretação.	Fornece uma explicação direta para as respostas obtidas pelo sistema (seqüência das regras aplicadas).

**Tab. 2.2 - Desvantagens do Conexionismo e Vantagens do Simbolismo**

Os sistemas híbridos neurosimbólicos implementam em módulos separados as abordagens: simbolista, conexionista e um módulo de integração neurosimbólica. A Fig. 2.11 mostra uma arquitetura híbrida neurosimbólica básica. A arquitetura híbrida neurosimbólica é definida por: um módulo que implementa uma estrutura simbólica (por exemplo, os sistemas

especialistas), um módulo que implementa uma estrutura conexionista (por exemplo, as redes neurais artificiais multicamadas) e um módulo de integração neurosimbólica [ORS 95B]. O módulo de integração neurosimbólica pode ser desenvolvido de várias formas a depender de fatores tais como: a forma como a integração será feita, o grau de acoplamento entre os módulos e o papel funcional que cada módulo executa em relação a cada um deles e ao ambiente externo. No capítulo 3 serão apresentadas algumas formas possíveis de implementação, funcionalidades e exemplos de uma arquitetura neurosimbólica baseadas na arquitetura da Fig. 2.11.



**Fig. 2.11 - Arquitetura Básica Híbrida**

No próximo capítulo apresenta-se um estudo dos principais sistemas híbridos neurosimbólicos, incluindo as suas classificações e arquiteturas.

## Capítulo 3

### SISTEMAS HÍBRIDOS NEUROSIMBÓLICOS

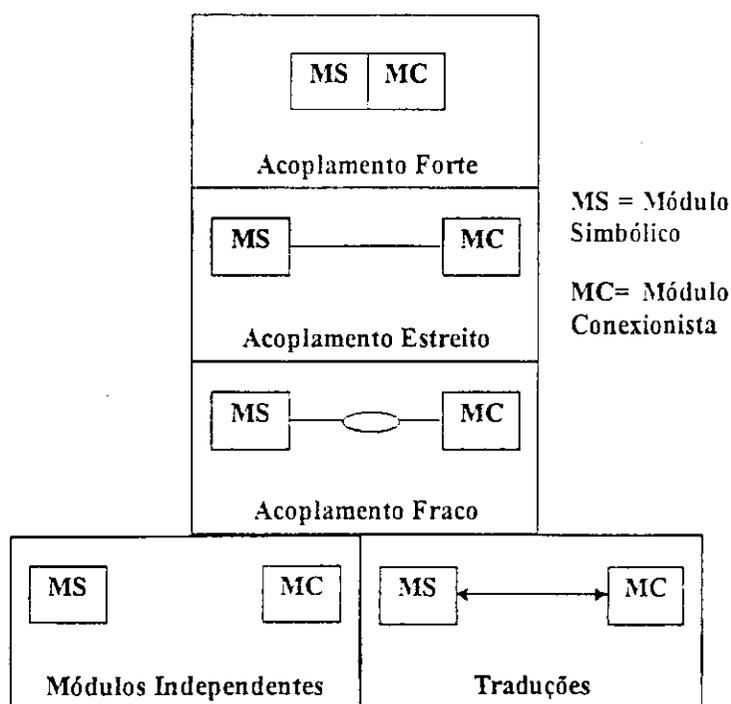
As primeiras pesquisas feitas na área de sistemas híbridos neurosimbólicos surgiram aproximadamente na metade da década de 80 [HIL 95]. Muitos estudos e alguns experimentos foram feitos para tornar a integração das abordagens simbolista e conexionista mais clara, propiciar futuros trabalhos e desenvolver novas técnicas. Este capítulo apresenta as classificações e arquiteturas dos sistemas híbridos neurosimbólicos e exemplos de sistemas existentes classificados conforme sua arquitetura e aplicação.

#### 3.1 Classificações dos Sistemas Híbridos Neurosimbólicos

Existe atualmente uma grande variedade de abordagens que tentam classificar os sistemas híbridos neurosimbólicos conforme alguns critérios e definições de cada classe de forma clara, e sem ambigüidades. Um problema comum é a identificação de um mesmo sistema híbrido neurosimbólico em mais de uma classe. [ORS95A].

A primeira classificação dos sistemas híbridos neurosimbólicos foi feita em 1992 por Medsker e Bailey (citados em [ORS95A]) e depois melhorada por Giacometti [GIA92]. Medsker e Bailey classificam os diversos tipos de sistemas conforme o grau de acoplamento dos módulos simbólico e conexionista (Fig. 3.1). Conforme esta classificação os módulos podem ser: fortemente, estreitamente ou fracamente acoplados. Os meios usados para a comunicação

entre os módulos é que determinam a sua classificação (Tab. 3.1).



**Fig. 3.1 - Classificação dos Sistemas Neurosimbólicos segundo Medsker e Bailey**

Esta classificação também apresenta a classe em que os módulos são completamente independentes e a classe de tradução. A tradução é a transferência de conhecimento, especificados através de regras, do módulo simbólico para o módulo conexionista, ou a extração de regras a partir da rede neural, transferindo conhecimento do módulo conexionista para o módulo simbólico.

Forma do Acoplamento	Meio de Comunicação
Forte	Procedimentos e Estruturas de Dados - interação contínua entre os módulos.
Estreito	Memória - relação de trocas bidirecionais de informação
Fraco	Arquivos - relação de entrada/saída trocas de informações numa única direção

**Tab.3. 1 - Classificação dos Sistemas Neurosimbólicos conforme o Meio de Comunicação**

A classificação dada por [SUN94] identifica quatro abordagens diferentes de sistemas híbridos neurosimbólicos, descritas abaixo:

1. *Abordagem Localista* - Agrupa todos os sistemas compostos unicamente de redes

neurais, onde cada neurônio representa um conceito simbólico. É constituída por redes neurais especializadas em emular o comportamento realizado por sistemas baseados na abordagem simbólica.

2. *Abordagem Distribuída* - Agrupa todos os sistemas compostos por redes neurais implementadas sobre uma representação distribuída, um conceito é codificado em vários neurônios [SUN93]. Possuem a mesma finalidade que a abordagem localista.
3. *Abordagem Local/Distribuída* - Agrupa os sistemas compostos de dois níveis integrados. O primeiro é baseado na abordagem localista e o segundo é baseado na abordagem distribuída.
4. *Outras Abordagens* - Todas as abordagens que não podem ser classificadas em nenhuma das três acima.

Outra classificação é apresentada por [HIL94], que distingue duas principais abordagens: a da união e a híbrida.

1. *Abordagem da união*- mais conhecida como sistema especialista conexionista, tem como propósito implementar estruturas simbólicas usando apenas redes neurais. É uma abordagem híbrida porque integra formas de representação do conhecimento simbólico em estruturas conexionistas(local/distribuída)[HIL 95]. Os sistemas desenvolvidos conforme esta abordagem, possuem as características de sistemas conexionistas que processam símbolos[HIL94]. Esta abordagem agrupa as arquiteturas localista, distribuída ou local/distribuída dependendo da forma como será representado o conhecimento.
2. *Abordagem híbrida*- os componentes simbólico e conexionista são completamente e individualmente desenvolvidos e possuem um ambiente de integração. Hilário em [HIL 94] diferencia quatro tipo de arquiteturas híbridas segundo o critério do modo de integração. São elas: pré/pós processamento, subprocessamento, metaprocessamento e cooperação.

Considera-se que a classificação mais completa dos sistemas híbridos neurosimbólicos, foi proposta por [ORS95A]. Ele apresenta uma hierarquia com três grandes classes, sob o ponto de vista do modo de integração e do grau de acoplamento (Fig. 3.2).

1. *Abordagem Híbrida* - Agrupa as abordagens que são fortemente, estreitamente e fracamente integradas.

2. *Abordagem Puramente Conexionista* - Agrupa as arquiteturas Localista, Distribuída e Local/Distribuída.
3. *Abordagem Semi-Híbrida* - Agrupa as abordagens que realizam as traduções, através da transferência de conhecimento de um módulo para o outro.

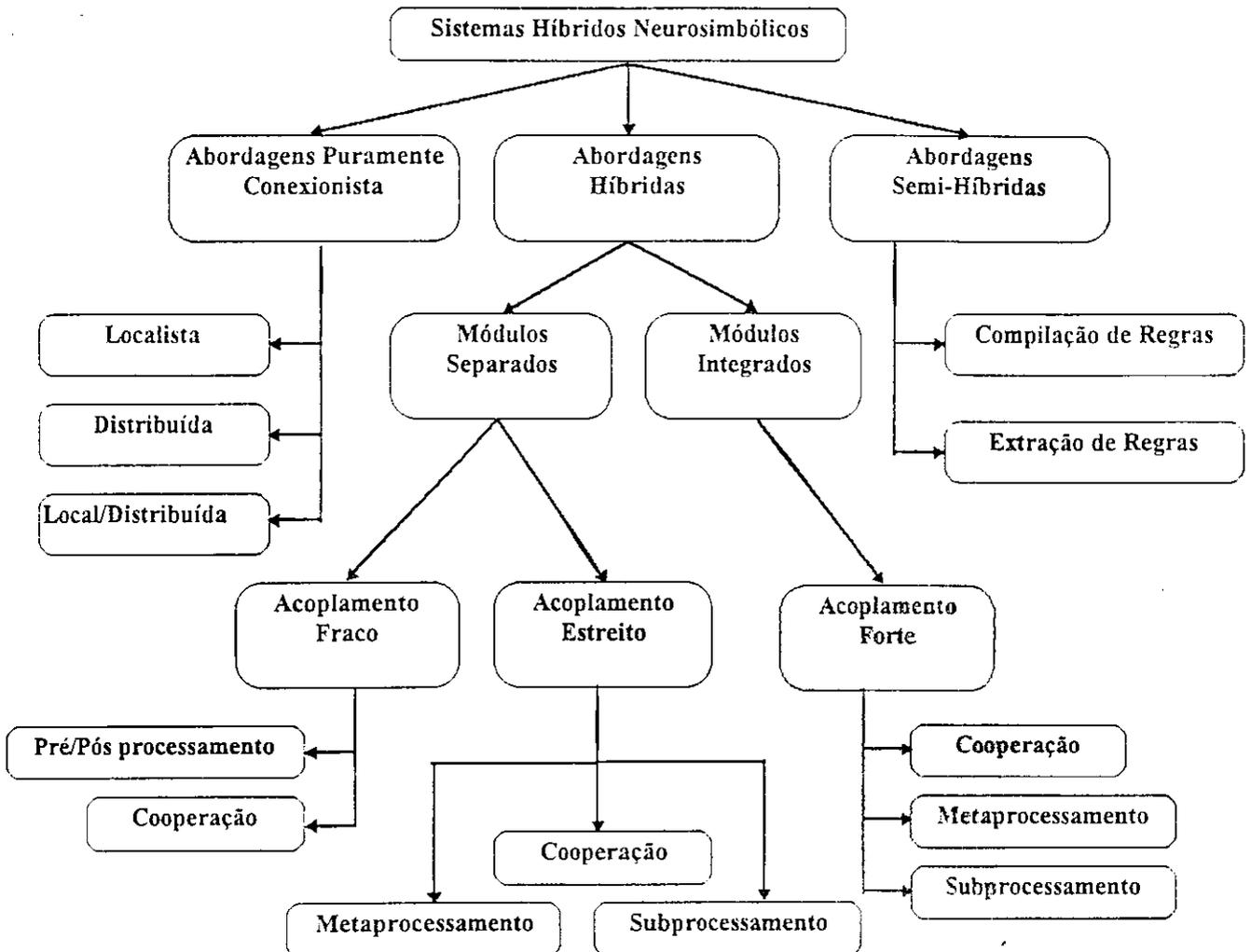


Fig. 3.2 - Classificação dos Sistemas Neurosimbólicos segundo Orsier

### 3.2 Arquiteturas dos Sistemas Híbridos Neurosimbólicos

Nas próximas seções serão apresentadas algumas arquiteturas que são utilizadas para o desenvolvimento de sistemas híbridos neurosimbólicos. Os exemplos de sistemas híbridos neurosimbólicos que serão citados podem ser classificados de formas diferentes a depender do critério de classificação utilizado.

#### 3.2.1 Arquitetura Híbrida Local/Distribuída

O Raciocínio Baseado em Casos (RBC) também faz parte do simbolismo e utiliza a similaridade de situações [SUN94]. O raciocínio baseado em casos é a essência de como o raciocínio humano funciona, porém, assim como os casos, as regras também são fundamentais, principalmente no momento da tomada de decisão. Somente quando não existem regras que possam ser aplicadas ou não existem conclusões baseadas em regras é mais prático aplicar o raciocínio baseado em casos. A arquitetura local/distribuída, como mostra a Fig.3.3, é composta de dois níveis que estão integrados. O primeiro é o local com o raciocínio baseado em regras e o segundo é o distribuído com o raciocínio baseado em casos.

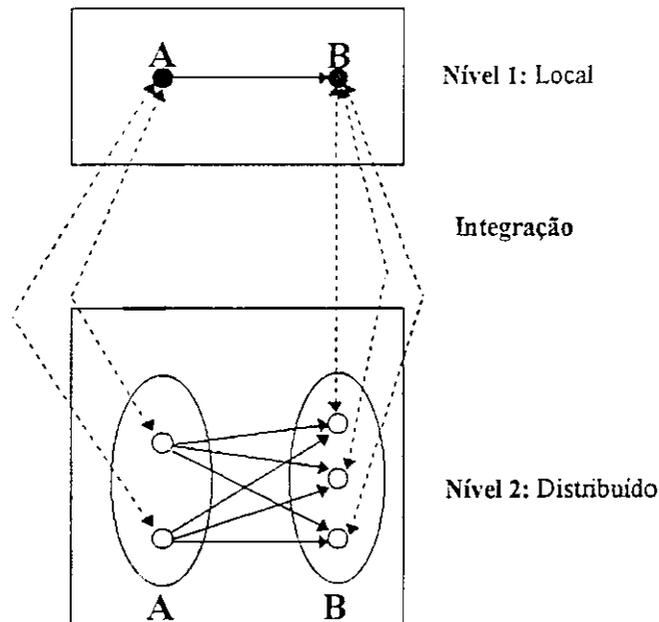


Fig. 3.3 - Arquitetura Local/Distribuída

No primeiro nível cada conceito ou proposição é representado por um neurônio da rede neural e as regras são implementadas através das conexões, interligando as condições e as conclusões. Neste nível, o raciocínio pode ser feito através de uma computação local. O segundo nível representa a parte distribuída, onde um conceito é codificado através de vários neurônios da rede neural, e cada neurônio possui uma parte da representação de cada conceito. Neste nível é implementado o raciocínio baseado em casos utilizando o grau de similaridade entre o conjunto de neurônios que representam dois conceitos. As conexões feitas no nível dois são conseqüências das conexões do nível um, ou seja, se existe uma conexão entre os neurônios A e B no nível um, então serão adicionadas no nível dois, conexões entre cada componente do conjunto que representa o conceito A com o conjunto que representa o conceito B. Esta forma

de implementação realiza a integração entre o conhecimento analítico, através das regras, e o intuitivo que pode ser parcialmente capturado através das redes neurais baseadas em similaridade de casos [SUN94].

### 3.2.2 Arquitetura Pré/Pós Processamento

Nesta arquitetura, (Fig. 3.4) os módulos que implementam as abordagens simbolista e conexionista são executados em seqüência. Um dos módulos é o principal resolvidor do problema e o outro módulo executa a tarefa de preparar os dados de entrada para o processamento principal, servindo como um pré-processamento, ou executa a tarefa de preparar os dados de saída para o ambiente externo, servindo como um pós-processamento [HIL95].

Os módulos são fracamente acoplados porque a integração está limitada à simples transferência de controle e informação de um módulo para o outro. A comunicação com o ambiente externo é feita principalmente pelo módulo que executa o pré ou pós-processamento. Algumas formas possíveis de configuração desta arquitetura são:

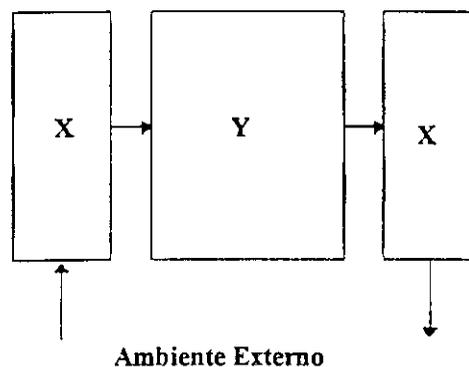


Fig. 3.4 - Arquitetura Pré/Pós Processamento

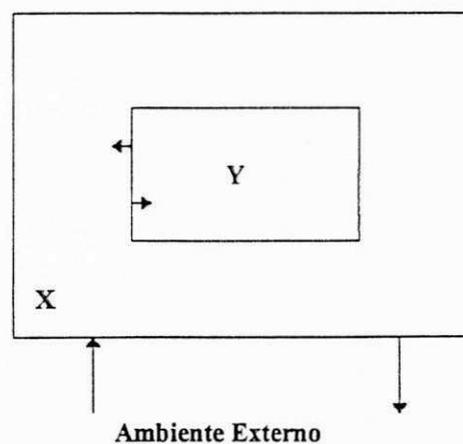
1. O módulo simbolista (Y) é o principal resolvidor do problema e é auxiliado pelo módulo conexionista (X) no pré ou pós-processamento.
2. O módulo conexionista (Y) age como o principal resolvidor do problema e é auxiliado pelo módulo simbolista (X) no pré ou pós-processamento.

Exemplos de sistemas que implementam este tipo de arquitetura são comentados em [HIL94] [HIL95]. No sistema ITRULE, que implementa uma arquitetura com a configuração 2, o pré-processador simbólico gera regras probabilísticas dos dados de entrada. As regras geradas são da forma *IF* <atributo> *ENTÃO* <valor> {com probabilidade p}. Todos os atributos são biná-

rios e podem ser diretamente mapeados numa rede neural. Os atributos condicionais das regras são mapeados como unidades de entradas, os atributos finais são mapeados como unidades de saída e os atributos métricos, como a probabilidade  $p$ , são mapeados como os pesos das conexões. A saída do pré-processador simbólico é carregada no processador principal, o simulador neural, que cria a rede neural correspondente às regras geradas no pré-processador simbólico. A configuração 1 é implementada pelo sistema SETHEO [HL 94], um provador seqüencial de teoremas de primeira ordem. Neste sistema, uma rede neural é treinada para agir como pré-processador e depois transfere seu resultado para o módulo simbólico. A tarefa do pré-processador conexionista é calcular a menor profundidade da rede para provar o teorema. Outro exemplo de implementação é quando o processador principal conexionista necessita explicar a estratégia de raciocínio para um usuário humano. Após o processamento principal conexionista, um pós-processador simbólico pode ser usado para gerar uma explicação.

### 3.2.3 Arquitetura Subprocessamento

Nesta arquitetura (Fig. 3.5), um dos módulos (Y) está embutido e subordinado ao outro módulo (X) que é o principal resolvedor do problema. O módulo subordinado é uma subrotina que é executada sob o controle de módulo principal.



**Fig. 3.5 - Arquitetura Subprocessamento**

Nesta arquitetura os módulos podem estar fortemente acoplados quando existe uma conexão direta entre eles que não está limitada a uma simples transferência de informação e controle como ocorre no pré/pós processamento. Por exemplo, uma rede semântica pode estar

conectada e corresponder a uma rede neural treinada para reconhecer micropropriedades<sup>11</sup>. O resultado da rede neural é processado pela rede semântica que determina os próximos passos para o resultado final. A comunicação com o ambiente externo é feita somente através do módulo principal.

Um exemplo deste tipo de arquitetura é a implementação de estruturas simbólicas no módulo principal e estruturas conexionistas no módulo subordinado, ou seja, um simulador neural executa tarefas específicas como parte da ação de uma regra que compõe o módulo simbólico. O sistema híbrido neurosimbólico desenvolvido neste trabalho, o SISNES (SIStema NEurosimbólico usando Subprocessamento) descrito no capítulo 4, foi implementado desta forma. Outro exemplo de implementação é quando uma regra é selecionada e uma rede neural associada a ela é ativada para computar fatores de incerteza dos fatos da regra [HIL 94].

### 3.2.4 Arquitetura Metaprocessamento

Nesta arquitetura, (Fig.3.6) um dos módulos implementa o processamento básico do problema, nível básico (X), e o outro módulo implementa o planejamento e acompanhamento da resolução do problema, nível meta (Y). São tarefas do nível meta: monitorar, controlar ou desenvolver performance do nível básico com o objetivo principal de orientar o desenvolvimento de tarefas complexas. O nível básico executa as instruções passadas pelo nível meta [HIL 95].

No metaprocessamento, assim como no subprocessamento, os módulos são fortemente acoplados existindo entre eles uma conexão direta de constante controle a cada passo do processo de resolução do problema. A comunicação com o ambiente externo é feita exclusivamente pelo nível meta, que é o responsável pelo controle das entradas e saídas das informações.

Esta arquitetura implementa o módulo conexionista como nível meta, que impõe controle de pesquisa para o módulo simbólico, que é implementado como nível básico. Por exemplo, uma rede neural pode guiar um resolvidor baseado em regras no processo de resolução de problemas. Em [HIL 94] é referenciado um sistema que resolve problemas de física mecânica utilizando esta arquitetura. Neste sistema, a rede neural inicialmente recebe como entrada as variáveis conhecidas e a variável objetivo e calcula a saída que será a próxima variável sub-

---

<sup>11</sup> Propriedades que isoladas têm pequenos significados mas que conjuntamente, definem um conceito.

objetivo que deverá ser calculada pelo módulo básico simbólico para possibilitar a solução da variável objetivo.

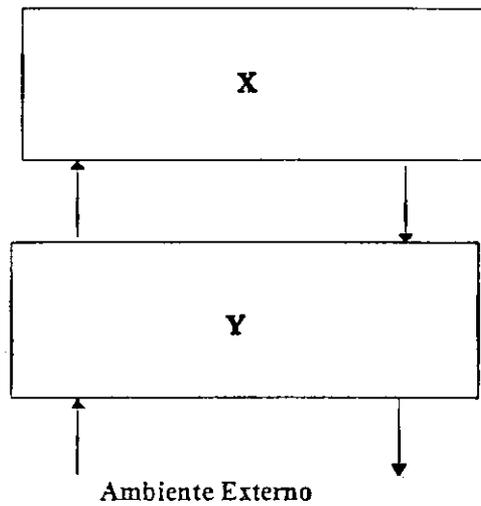


Fig. 3.6 - Arquitetura Metaprocessamento

Esta arquitetura também pode ser usada para provar teoremas de primeira ordem, de uma forma diferente da arquitetura pré/pós processamento. O controle da pesquisa, para o melhor caminho, é feito pelo nível meta, através de estruturas conexionistas. Métodos heurísticos são usados para evitar a explosão combinatória e escolher a alternativa mais promissora na prova de teorema. O nível básico implementa estruturas simbólicas e executa as escolhas feitas no nível meta.

### 3.2.5 Arquitetura Cooperação

Nesta arquitetura (Fig. 3.7) os módulos são igualmente parceiros no processo de solução do problema, não existe subordinação ou controle de um módulo sobre o outro. Cada módulo pode transmitir e receber informações um do outro. O módulo simbólico transmite informações para o módulo conexionista através da compilação de estruturas simbólicas em estruturas conexionistas, por exemplo, a compilação de regras numa rede neural. O módulo conexionista transmite informações para o módulo simbolista através da extração de estruturas simbólicas das estruturas conexionistas, por exemplo, a extração de regras de uma rede neural.

Os módulos estão fracamente acoplados, a integração está limitada somente à transferência de informações de um módulo para o outro. A comunicação com o ambiente externo é

feita diretamente por cada módulo e de forma independente.

Um exemplo de implementação desta arquitetura é o SYNHESYS [GIA92], utilizado no desenvolvimento de sistemas híbridos neurosimbólicos, com característica incremental, onde ambos os módulos conexionista e simbolista podem ser usados na aquisição do conhecimento e também na solução do problema, através da interpretação de regras e ativação da rede neural.

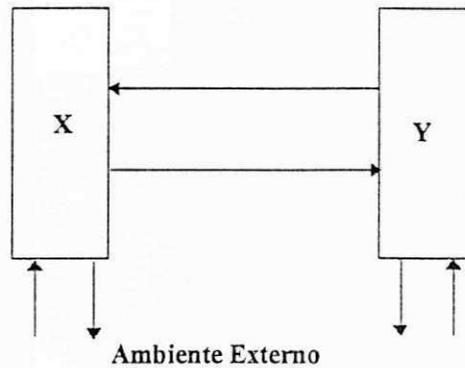


Fig. 3.7 - Arquitetura de Cooperação

### 3.3 Exemplos de Sistemas Híbridos Neurosimbólicos

Nesta seção serão descritos alguns exemplos de sistemas híbridos neurosimbólicos, suas arquiteturas e aplicações.

#### 3.3.1 KBANN

KBANN-Knowledge-Based Artificial Neural Networks é um sistema híbrido desenvolvido por [TOW91A]. O KBANN adquire conhecimentos através da combinação de regras simbólicas e exemplos. Testes feitos utilizando o KBANN, na área de biologia molecular e no estudo do raciocínio geométrico em crianças, comprovaram a sua eficiência na capacidade de aprender através de regras e exemplos [TOW90][TOW91B].

A aquisição de conhecimentos no KBANN é dividida em três etapas e executa três algoritmos independentes: o tradutor de regras-rede, o refinador e o tradutor de rede-regras (Fig. 3.8). Estes três algoritmos, quando usados em conjunto, formam um sistema genérico de aquisição de conhecimento. As três etapas são:

#### **Etapa 1 - Inserir regras simbólicas em uma rede neural:**

Inicialmente o conhecimento do KBANN é composto por regras simbólicas corretas ou aproximadamente corretas que formam o domínio teórico inicial.(Fig. 3.8) Estas regras, através da execução do *tradutor regras-rede*, irão determinar a criação da topologia da rede neural inicial, os pesos e conexões iniciais. Desta forma, a rede neural inicial será capaz de gerar as mesmas conclusões que seriam geradas pelas regras do domínio teórico inicial. O tradutor de regras-rede é responsável por mapear a estrutura das regras em uma rede neural inicial e realizar a execução do compilador de regras (Tab. 3.2). A rede neural inicial, depois de criada, possui uma topologia multicamadas. O número de camadas e o número de neurônios em cada camada é determinado pelo conjunto de regras do domínio teórico inicial.

Sistema Especialista	Rede Neural
Conclusão Final	Unidades de Saída
Suposições e Fatos	Unidades de Entrada
Conclusões Intermediárias	Camadas Escondidas
Subordinações	Pesos das Conexões

**Tab. 3. 2 - Tabela de Mapeamento Regras-Rede do KBANN**

**Etapa 2 - Refinamento da Rede Neural Inicial:**

A segunda etapa realiza o refinamento da rede neural (criada na primeira etapa), utilizando o algoritmo de aprendizagem de retropropagação e um conjunto de exemplos (Fig. 3.8). Com a execução do refinamento, a rede neural terá condições de refinar os conhecimentos anteriormente aprendidos e realizar correções de eventuais erros do domínio teórico inicial do módulo simbólico como, por exemplo, dados imprecisos ou incompletos. Após a conclusão deste módulo, a rede pode ser usada para classificar com maior precisão os padrões apresentados.

**Etapa 3 - Extração de Regras Simbólicas da Rede Neural Refinada:**

Na terceira etapa, é realizada a extração de regras simbólicas da rede neural refinada na segunda etapa, disponibilizando uma explicação dos possíveis resultados gerados pela rede neural. Através deste processo, as informações contidas na rede neural, podem ser avaliadas ou revisadas por um especialista humano. O tradutor de rede-regras é responsável por extrair regras simbólicas de uma rede neural, através da execução de

um algoritmo específico. Esta parte do sistema apresenta três benefícios importantes para o aprendizado de máquinas:

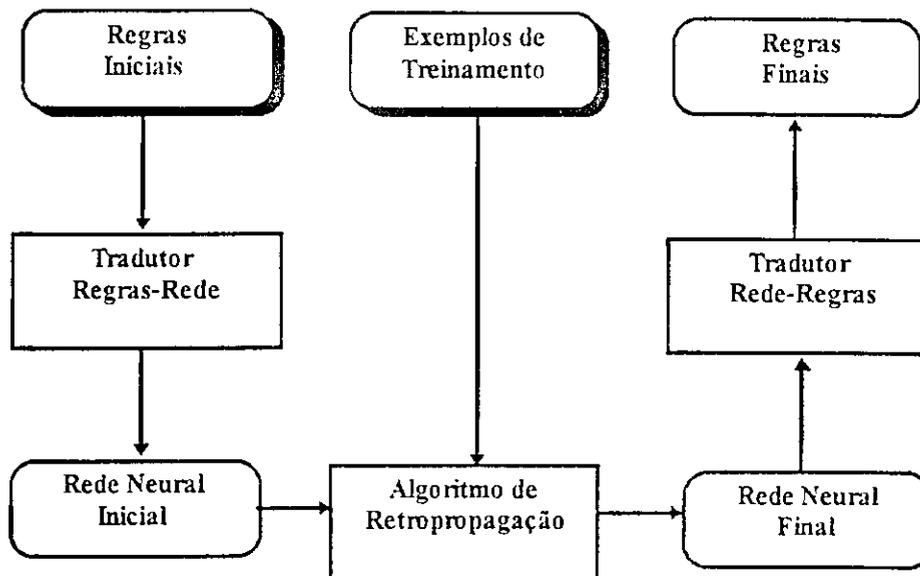


Fig. 3.8 - Arquitetura do KBANN

1. As regras extraídas estão disponíveis durante o treinamento para uma revisão humana;
2. As regras extraídas podem ser usadas para gerar uma explicação para os resultados da rede neural;
3. As novas regras aprendidas ou modificadas no refinamento, podem ser adicionadas no domínio teórico formando um novo conjunto de regras possivelmente mais correto. O módulo simbólico com as novas regras adicionadas, pode ser usado para criar uma nova rede neural, voltar a executar a primeira etapa, formando um ciclo (Fig.3.9).

Observando a forma como a rede neural inicial é construída no KBANN, algumas das desvantagens da abordagem conexionista são eliminadas ou significativamente reduzidas. Por exemplo, o problema das *características importantes dependem do contexto* e o problema das *características irrelevantes*, descritos na Seção 2.2 do Capítulo 2, são significativamente

reduzidos. Além disso, o KBANN permite que o domínio teórico inicial seja incompleto ou que não esteja correto. Esta falha será eliminada na segunda etapa.

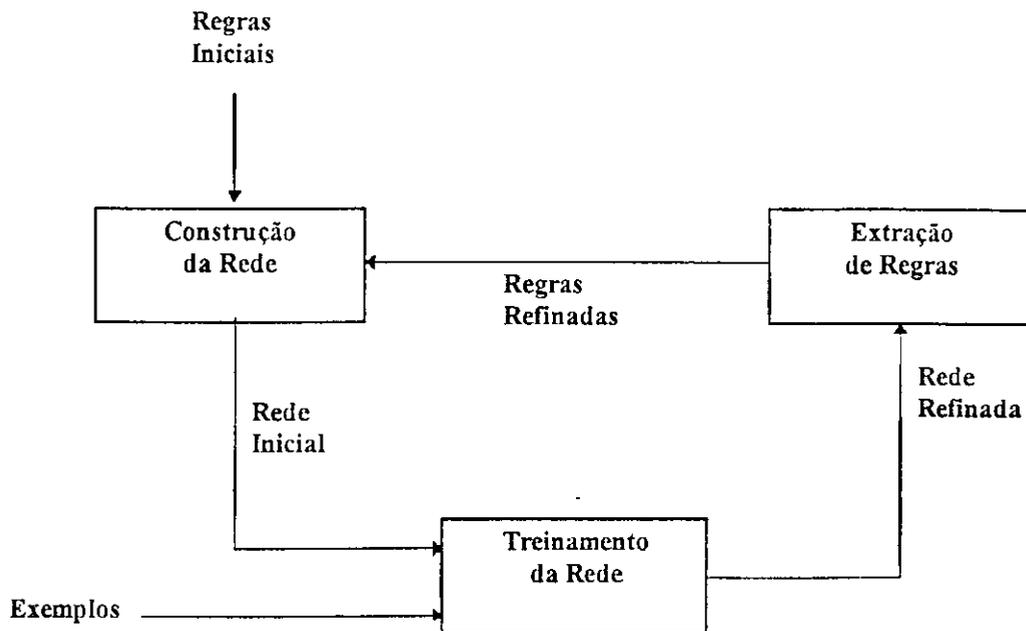


Fig. 3.9 - Ciclo do KBANN

### 3.3.2 Extensões do KBANN

Algumas evoluções foram adicionadas ao projeto inicial do KBANN com o objetivo de reduzir algumas falhas ou incluir melhorias[TOW91A]. Descreveremos a seguir duas evoluções: a primeira é um pré-processador simbólico o DAID - Desired Antecedent Identification, e a segunda evolução é um mecanismo de inclusão de camadas escondidas da rede neural inicial do KBANN [TOW91].

O objetivo do DAID é aperfeiçoar a capacidade de aprendizado do KBANN. O conjunto de regras iniciais, que compõem o domínio teórico inicial, não identifica todas as características de entrada importantes e necessárias para o aprendizado correto e para a perfeita criação da rede neural inicial. Alguns testes demonstraram que o KBANN ignora as características de algumas entradas consideradas inúteis. O DAID fornecerá informações para o KBANN sobre o potencial de utilidade de cada característica de entrada, não permitindo que sejam ignoradas e sim que sejam consideradas com um peso pequeno.

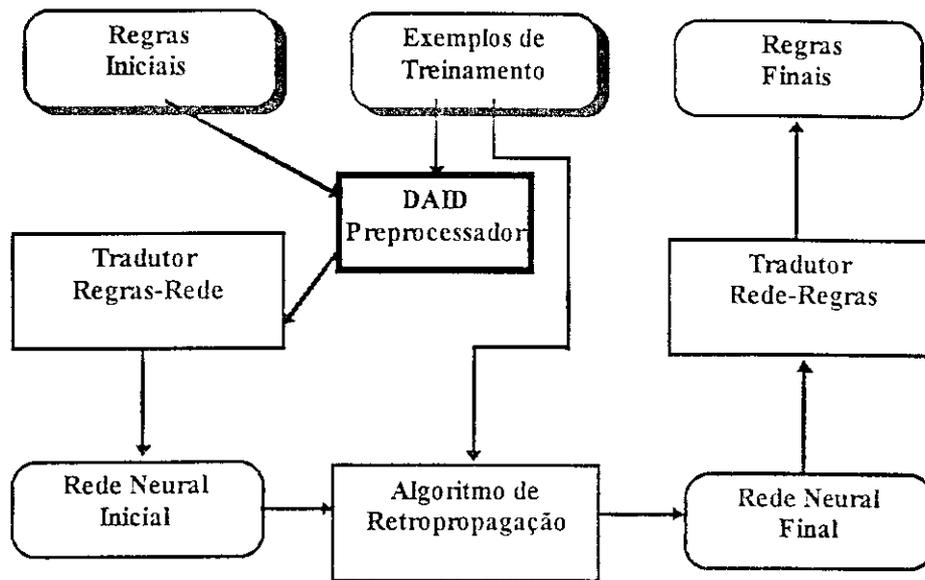


Fig. 3.10 - Arquitetura KBANN-DAID

A inclusão do DAID na arquitetura do KBANN (Fig. 3.10) adiciona o quarto algoritmo nos passos do KBANN, situado entre as regras iniciais fornecidas pelo usuário e o tradutor de regras-rede. O algoritmo simbólico do DAID recebe como entrada as regras iniciais e um conjunto de exemplos de treinamento para complementar as informações que serão fornecidas para o tradutor de regras-rede. A saída do tradutor de regras-rede não é simplesmente uma cópia das regras iniciais mas criará uma rede neural inicial mais eficiente.

A segunda evolução é um mecanismo de adição de camadas escondidas na rede neural do KBANN. O objetivo deste mecanismo é incluir vocabulário suficiente para precisão do aprendizado da rede neural do KBANN. Geralmente as regras iniciais não fornecem vocabulário suficiente para a criação de todas as camadas escondidas necessárias para o aprendizado preciso de um conceito. Neste caso, a rede irá falhar no processo de aprendizado ou será forçada a adaptar as camadas existentes para múltiplas necessidades. No primeiro caso, a rede não irá aprender a um nível aceitável. No segundo caso, os resultados não serão interpretados pela rede. Em ambos os casos, o adição de novas camadas escondidas será necessário para fornecer novo vocabulário e aperfeiçoar o aprendizado.

### 3.3.3 INSS

O INSS-Incremental Neuro-Symbolic System é um sistema híbrido desenvolvido no laboratório LIFIA (Laboratoire d'Informatique Fundamentale et d'Intelligence Artificielle) do INPG (Institut National Polytechnique de Grenoble) [OSO95]. Este sistema é um aperfeiçoamento do KBANN, com o objetivo de reduzir seus pontos fracos e introduzir algumas melhorias, como o uso do modelo de rede neural **Cascade-Correlation** [FAH90].

A arquitetura e o processo de aprendizagem do INSS é baseado na compilação de regras, na refinamento e na extração de regras do KBANN. A diferença está na escolha e implementação do modelo de rede neural artificial a ser adotado para o sistema. O KBANN adota o modelo de rede neural multicamadas com o algoritmo de aprendizagem de retropropagação [RUM89]. O INSS utiliza o modelo de rede neural Cascade-Correlation com o algoritmo de aprendizagem QUICKPROP [FAH90].

A topologia da rede neural é determinada pela compilação de regras baseadas no domínio teórico inicial do problema. Se o domínio teórico estiver incompleto surge a necessidade de expandir a rede neural, e uma das opções é utilizar um modelo de rede neural incremental, por exemplo, o cascade-correlation [OSO 95]. O algoritmo QUICKPROP é extremamente rápido e soluciona o problema de convergência, e utiliza a técnica de congelamento de pesos das unidades inseridas solucionando o problema de competição entre as unidades.

A arquitetura do INSS é composta por dois módulos o simbólico e o conexionista (Fig.3.11). Possui os processos de transferência e aperfeiçoamento de conhecimento através da compilação e extração de regras e refinamento da rede neural, similares ao KBANN. Evoluções no INSS estão concentradas nas potencialidades do módulo simbolista, permitindo regras mais elaboradas, com a inclusão de variáveis. Uma das possibilidades seria transformar o módulo simbolista num subsistema do módulo conexionista, passando da atual arquitetura híbrida de processamento corporativo (cooperação) para a arquitetura com processamento subordinado (subprocessamento).

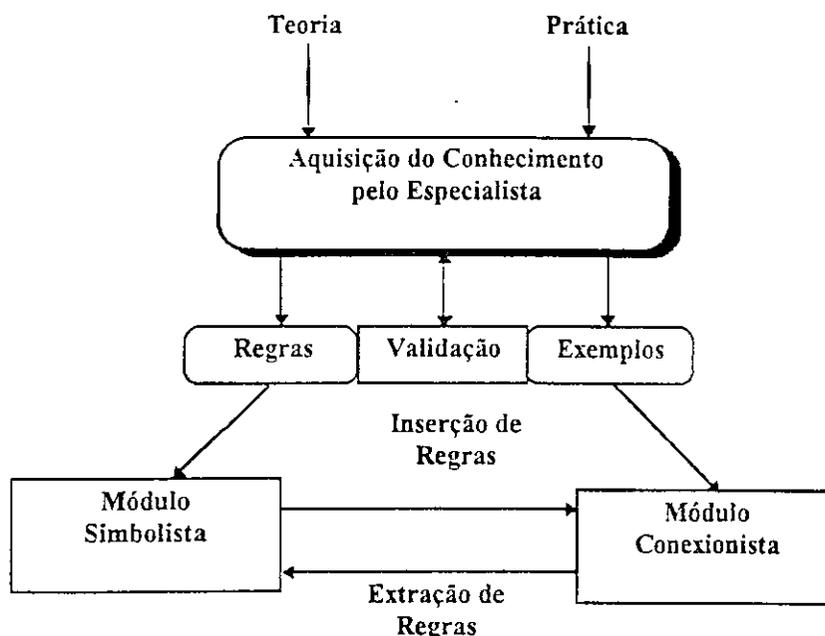


Fig. 3.11 - Arquitetura do INSS

### 3.3.4 SYNHESYS

O SYNHESYS-SYmbolic and Neural Hybrid Expert SYstem Shell é um sistema híbrido neurosimbólico incremental para desenvolvimento de sistemas inteligentes desenvolvido por [GIA92], e pode ser usado em diversas aplicações, como por exemplo, na área naval [ORS94].

O SYNHESYS, além dos módulos simbolista e conexionista, é composto por dois componentes importantes que possuem a função de integração desses módulos, são eles: o gerenciador de integração e o processo de transferência de conhecimento. É através desses dois componentes que existe a comunicação entre os módulos simbolista e conexionista, permitindo ao especialista do domínio interagir com o sistema híbrido (Fig. 3.12).

O gerenciador de integração é responsável pela determinação de uma decisão. Por exemplo, determinar a validação da resposta de um dos dois módulos, ou determinar que a resposta de um dos dois módulos seja a solução final do sistema híbrido. O processo de transferência de conhecimento é responsável pela compilação ou extração de regras em uma rede neural, transferindo conhecimento das regras para a rede neural, ou extraindo novas regras da rede neural e incluindo-as nas regras do módulo simbolista.

O módulo simbolista é desenvolvido como um sistema especialista baseado em regras com raciocínio para frente e para trás. As entradas do módulo simbolista são em forma de

situações. A fase de aquisição do conhecimento é feita através de um especialista do domínio e independente do módulo conexionista, porém, o módulo conexionista poderá incluir novas regras no módulo simbolista.

O módulo conexionista é composto por uma rede neural multicamadas baseada em protótipo, é incremental e supervisionada. Neste tipo de rede neural, cada entrada corresponde a um protótipo de entrada ou a uma situação de entrada. O número de camadas escondidas não é fixo, e durante o processo de treinamento (aprendizado) ou durante o processo de reconhecimento, novas camadas podem ser adicionadas, quando necessário. Esta é a característica incremental da rede. Este tipo de rede neural demonstra importantes propriedades: primeiro, a rede neural é do tipo local<sup>12</sup>, o que facilita sua transformação em regras; segundo, a característica incremental facilita o acréscimo do novos conhecimento.

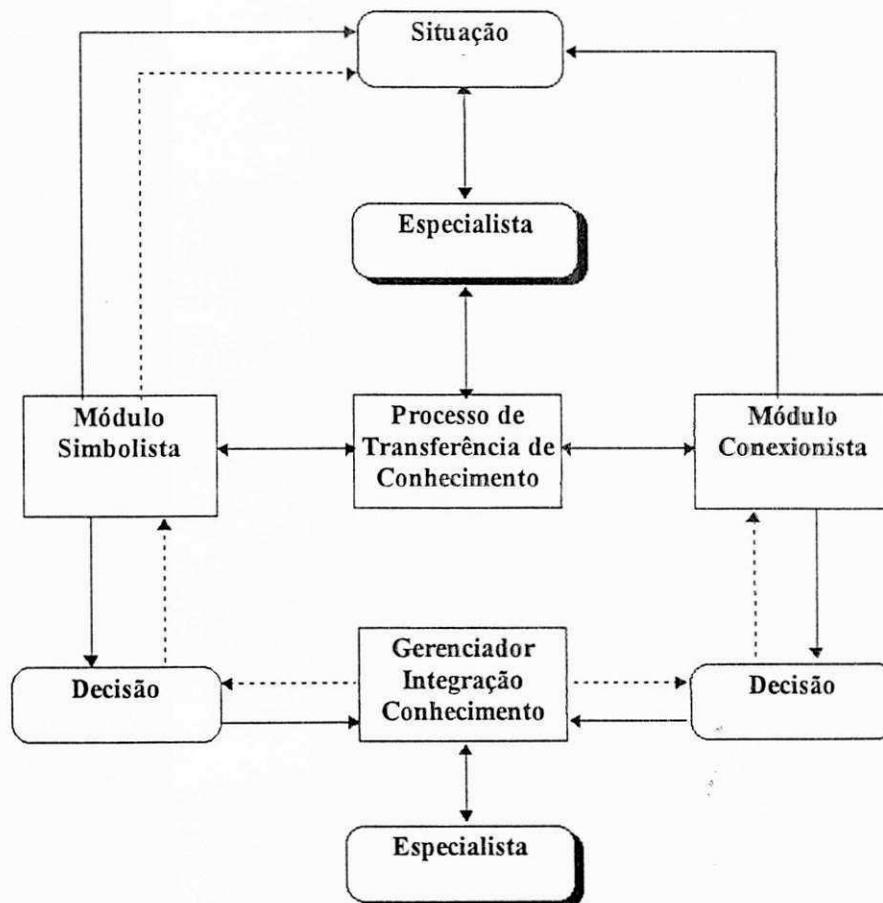


Fig. 3.12 - Arquitetura do SYNHESYS

A integração entre os módulos é executada pelo gerenciador de integração e obedece os

<sup>12</sup> Cada unidade da rede corresponde a um determinado protótipo ou situação.

passos descritos abaixo. Usaremos a seguinte legenda:

- MC → Módulo Conexionista
- MS → Módulo Simbólico
- SC → Solução Conexionista
- SS → Solução Simbólica
- SE → Solução do Especialista

**Passo 1 - Ativação do MC:**

As entradas são apresentadas ao MC.

Se MC for capaz de determinar uma solução SC então

*O gerenciador de integração solicita a validação pelo MS (Passo 3)*

Senão

*O gerenciador de integração solicita uma solução SS do MS (Passo 2)*

**Passo 2 - Ativação do MS:**

As entradas são apresentadas ao MS

Se MS for capaz de determinar uma solução SS via raciocínio para frente então

*O gerenciador de integração determina que SS é a solução final do sistema, e solicita ao especialista uma consulta de aprovação (Passo 5).*

Senão

*O gerenciador de integração determina que o sistema não pode processar uma solução para o problema(Passo 6).*

**Passo 3 - Validação do MS:**

Ao MS é solicitado uma validação da solução SC determinada pelo MC

Se MS for capaz de validar a solução SC via raciocínio para trás então

*O gerenciador de integração determina que SC é a solução final do sistema, que pode ser explicada por MS e solicita ao especialista uma consulta de aprovação (Passo 5).*

Senão

*O gerenciador de integração determina uma situação de conflito*

*(Passo 4).*

**Passo 4 - Situação de Conflito:**

As entradas são apresentadas ao MS, e é solicitada uma solução.

Se MS for capaz de determinar uma a solução SS via raciocínio para frente então

*A solução SS deve ser necessariamente diferente da SC. O gerenciador de integração determina uma situação de conflito que pode ser resolvida pelo especialista ou através de processos heurísticos.*

Senão

*O gerenciador de integração determina que SC seja a solução final do sistema, que não pode ser explicada por MS e solicita ao especialista uma consulta de aprovação (Passo 5).*

**Passo 5 - Consulta ao Especialista:**

O especialista poderá ter três comportamentos:

1. Concordar com a solução do sistema;
2. Não concordar com a solução do sistema, e determinar a solução correta SE. Neste caso, o MC deverá aprender uma nova associação para determinar SE ou o MS deverá incluir ou modificar suas regras.
3. Não concordar com a solução do sistema, e não conseguir determinar a solução correta. Neste caso, o MC deverá esquecer a associação que determinou SC.

**Passo 6 - Aquisição Incremental de Conhecimento:**

Neste caso, o sistema não é capaz de determinar uma solução. Então o gerenciador de integração indaga ao especialista se o MC deverá aprender uma nova associação.

### 3.4 Compilação e Extração de Regras

Em alguns sistemas híbridos neurosimbólicos, a transferência de conhecimento do módulo simbólico para o módulo conexionista é feita através da compilação de regras, que geralmente resulta na elaboração de uma rede neural inicial. Após esta etapa a rede neural inicial está pronta para o refinamento ou aperfeiçoamento dos conhecimentos adquiridos. A transferência

de conhecimento do módulo conexionista para o módulo simbólico é feita através da extração de regras. Após a etapa do refinamento a rede neural está com novos conhecimentos adquiridos dos eventuais erros das regras iniciais. As novas regras extraídas, que foram adquiridas após o refinamento, podem ser incluídas no conjunto de regras iniciais formando um ciclo e possibilitando a criação de explicações para as soluções encontradas.

Esta seção apresenta a compilação e extração de regras, utilizado em sistemas híbridos neurosimbólicos, como o KBANN e o INSS.

### 3.4.1 Compilação de Regras

A compilação de regras parte de um conjunto de regras do cálculo proposicional de ordem 0<sup>+</sup>

As regras são representadas da seguinte forma:

*<conseqüente> :- <antecedente1> [, <antecedente2>, ...] onde;*

- **<conseqüente>** é um predicado que pode assumir os valores V ou F, de acordo com a avaliação dos seus predicados antecedentes;
- **<antecedente>** é um predicado (com valores discretos ou contínuos), podendo ser precedido do operador lógico de negação NOT para valores discretos;

Os operadores lógicos são:

- **OU (Disjunção)** - Um conjunto de duas ou mais regras formam uma disjunção;
- **E (Conjunção)** - Um conjunto de dois ou mais antecedentes em uma mesma regra formam uma conjunção;
- **NEGACÃO** - Predicados, do tipo binário, negados.

Algumas restrições das regras são necessárias para que seja possível a realização da compilação sob a forma de uma rede neural, que funcionalmente será equivalente ao comportamento do módulo simbolista que manipule tais regras. Estas restrições podem exigir um pré-processamento para a transformação do conjunto de regras iniciais em um conjunto de regras que seja semelhante a hierarquia de uma rede neural. As restrições impostas são mostradas na primeira etapa do processo de compilação de regras.

O processo de compilação é realizado através das seguintes etapas iniciais.

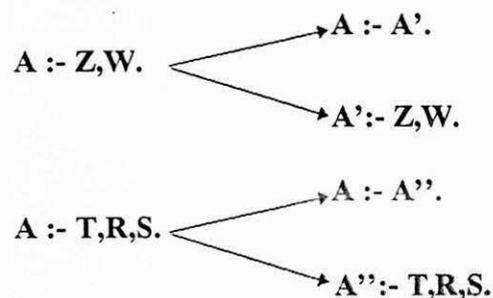
### **Etapas 1 - Rescrever Regras**

Nesta etapa todas as regras são avaliadas e serão re-escritas as regras que possuem o mesmo conseqüente e mais de um antecedente. As disjunções só podem ter um único antecedente, a disjunção abaixo será re-escrita.

$A :- Z, W.$

$A :- T, R, S.$

Neste caso, as regras serão re-escritas e transformadas cada uma em duas outras regras. A primeira terá o conseqüente original e um antecedente novo, a segunda terá como conseqüente o antecedente novo criado na regra anterior e como antecedente os antecedentes originais da regra. No exemplo dado as regras seriam re-escritas da seguinte forma:



As regras que são re-escritas não afetam o significado do conjunto de regras. Em [TOW 91B] são mostrados teoremas e provas que garantem este processo. As regras adicionadas facilitam e deixam quase imediata a transformação para a hierarquia das redes neurais (Fig. 3.13).

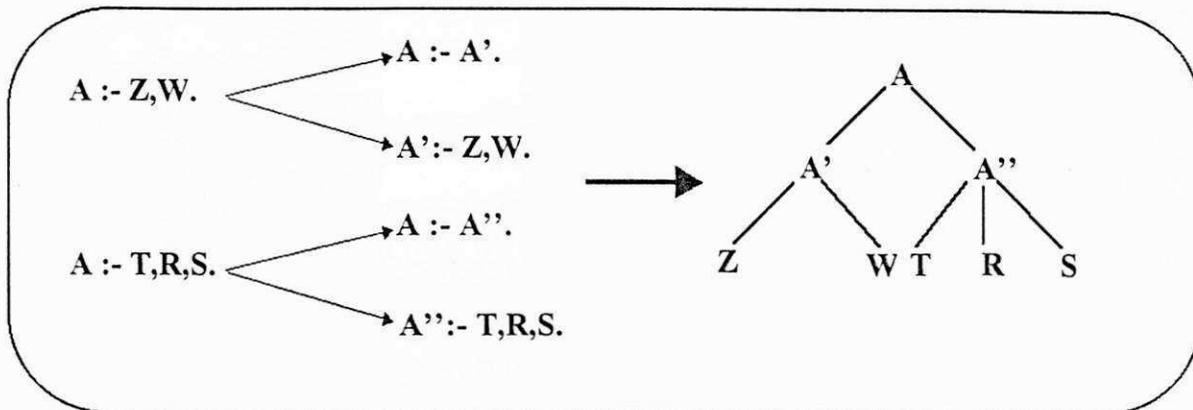


Fig. 3.13 - Transformação das Regras para a hierarquia das Redes Neurais

### Etapa 2 - Mapeamento das Regras

Nesta etapa é feito o mapeamento do conjunto de regras da etapa 1 em uma rede neural. Este mapeamento cria uma rede neural com uma correspondência de um para um com os elementos das regras. As regras com antecedentes positivos possuem conexões correspondentes à rede neural com pesos iguais a  $+W$ , e regras com antecedentes negativos têm conexões com pesos iguais a  $-W$ .  $W$  é um valor constante (cada sistema atribuiu um valor diferente). No KBANN as disjunções têm o deslocamento (BIAS) do neurônio ajustado para  $\text{Bias} = -0.5 * W$ . As conjunções têm o Bias do neurônio ajustado para  $\text{Bias} = (-P + 0.5) * W$ .  $P$  é o número de antecedentes positivos numa dada regra.

No final desta etapa, a rede neural inicial estará criada e possui as mesmas informações que o conjunto de regras. Outras etapas podem ser necessárias para garantir o melhor aprendizado da rede neural e facilitar a correção de possíveis erros. As etapas adicionais são para criar novas conexões, adicionar novos neurônios na camada de entrada e adicionar camadas escondidas.

### 3.4.2 Extração de Regras

Os algoritmos para a extração de regras mais utilizados são:

#### Algoritmo SUBSET

Este algoritmo tem por objetivo encontrar subconjuntos de neurônios tais que a ativação conjunta de todos eles irá ativar o neurônio subsequente. Este algoritmo é descrito e provado a sua eficiência nos seguintes trabalhos [OS095][TOW91A]. A Fig.3.14 mostra um subconjunto de neurônios necessários para ativar o neurônio subsequente e suas respectivas regras.

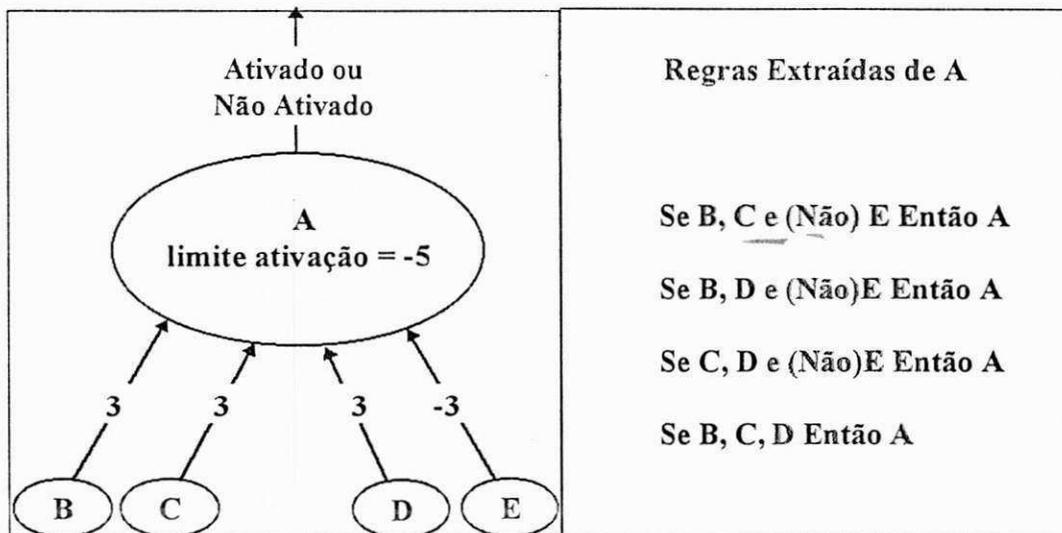


Fig. 3.14 - Regras Extraídas da Ativação de um Neurônio

Este algoritmo extrai um grande número de regras possíveis, esta é a sua desvantagem. Estas regras podem ser simplificadas ou representadas de forma diferente com o objetivo de diminuir a quantidade de regras que deverão ser tratadas.

As regras são formadas quando a soma dos pesos das conexões dos neurônios ultrapassa o limite de ativação. No exemplo da Fig. 3.14 quatro regras foram encontradas, a soma dos pesos das conexões ultrapassaram o limite de ativação -5, as regras extraídas da rede neural podem ser simplificadas na regra abaixo.

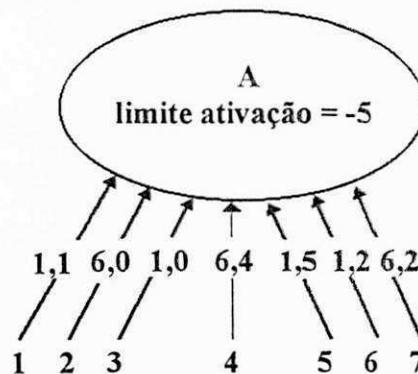
***Se um conjunto com 3 elementos de {B, C, D, Não(E)} Então ativar A***

### Algoritmo NofM

O algoritmo NofM foi proposto com o objetivo de minimizar a deficiência do algoritmo SUBSET, em extrair um grande número de regras. A idéia básica deste algoritmo é que as conexões isoladas não representam muita importância na criação das regras, e sim grupos com conexões similares. Este algoritmo pesquisa quantos  $N$  dos  $M$  antecedentes de uma regra são verdadeiros.

O primeiro passo deste algoritmo é separar em grupos as conexões com pesos iguais ou similares. O segundo passo é calcular a média dos componentes de cada grupo. O terceiro passo é eliminar os grupos em que a soma da média combinados com os componentes do mesmo grupo e dos outros grupos não conseguem ultrapassar o limite de ativação. O quarto passo é extrair e simplificar as regras que ultrapassam o limite de ativação. As regras são do tipo:

*Se  $N$  das  $\{M$  conexões são verdadeiras} então a regra é verdadeira*



**Fig. 3.15 - Neurônio com seus Pesos e Conexões**

No exemplo da Fig. 3.15 inicialmente existem sete conexões com pesos diferentes, após a execução do primeiro passo (Fig. 3.16) são criados dois grupos com as médias 6,1 e 1,1.

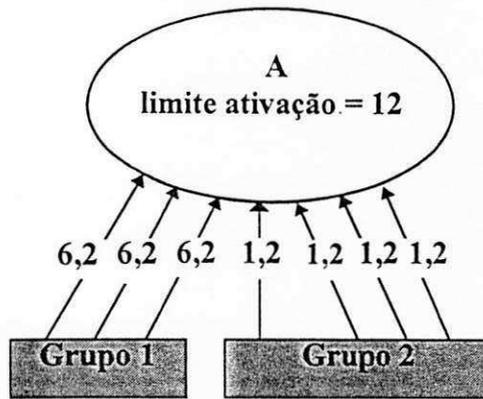


Fig. 3.16 - Neurônio com seus Pesos e Conexões

No terceiro passo o grupo 2 é eliminado, porque não existem uma combinação com este grupo que ultrapasse o limite de ativação (Fig. 3.17).

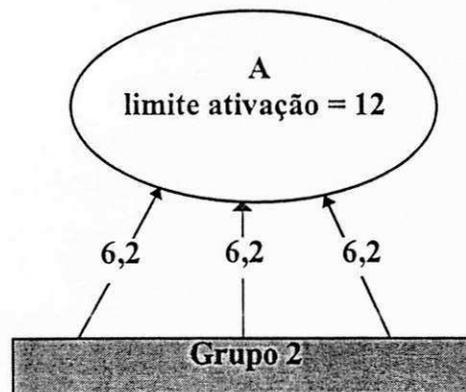


Fig. 3.17 - Neurônio com seus Pesos e Conexões

No quarto passo observa-se que se dois componentes do grupo 2 forem verdadeiros então a saída da função que o neurônio implementa será maior que 12, o limite de ativação, e o neurônio será ativado. Algumas regras podem ser extraídas e simplificadas na seguinte regra:

*Se 2 do {grupo 2} então A*

No próximo capítulo será descrito o sistema híbrido neurosimbólico proposto, denominado SISNES, sua arquitetura e classificação.

## Capítulo 4

### O SISNES

Neste capítulo será descrito o SISNES - SISTema NEurosimbólico usando Subprocessamento - a sua arquitetura e classificação. Serão apresentados os módulos que compõem o sistema e sua implementação.

#### 4.1 A Arquitetura do SISNES

O SISNES foi desenvolvido e implementado num microcomputador tipo IBM-PC, com a arquitetura híbrida neurosimbólica de subprocessamento, como mostrado na seção 3.2.3 do Capítulo 3. Nessa arquitetura, o módulo connexionista está embutido e subordinado ao módulo simbolista, que é o principal resolvidor do problema. O módulo subordinado é executado sob o controle de módulo principal, como mostra a Fig.4.1.

O módulo simbolista do SISNES implementa o domínio teórico através de regras que são implementadas conforme o LPA-Prolog<sup>13</sup>. O módulo connexionista do SISNES é responsável pela determinação de informações que são melhor obtidas por redes neurais. Este módulo é composto por redes neurais do tipo multi camadas, com neurônios do tipo perceptrons (ver Seção 2.2.2), e treinada com o algoritmo de retropropagação (ver Seção 2.2.3). O módulo connexionista do SISNES é implementado em Borland C++ (DOS Standard).

---

<sup>13</sup> O LPA-Prolog foi a ferramenta escolhida para o desenvolvimento do módulo simbólico principalmente pela sua facilidade de comunicação com outras linguagens de programação.

A integração neurosimbólica, comunicação entre o módulo conexionista e simbólico, é feita através da memória principal e de arquivos em disco (Fig.4.2). No SISNES, quando é necessário ativar o módulo conexionista (rede neural), o módulo simbolista envia os dados de entrada da rede neural e aciona o módulo conexionista.

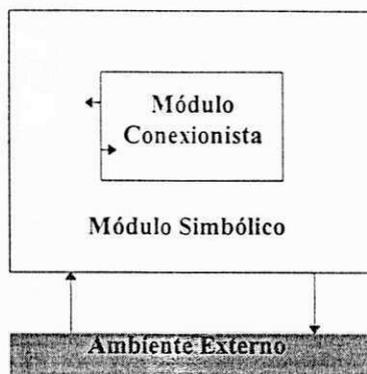


Fig. 4.1 - Arquitetura de Subprocessamento do SISNES

O LPA-Prolog permite que seja executado um programa executável através do comando chamado: EXEC(“Caminho do programa”), é basicamente através deste comando que o módulo simbolista aciona o programa que implementa a rede neural, o módulo conexionista. Enquanto o módulo conexionista está executando o módulo simbolista fica esperando, numa malha de espera ou *loop* a sinalização de término. A sinalização é feita também através de passagens de parâmetros utilizando a memória e arquivos (Fig. 4.2). Quando o módulo conexionista termina, ele cria um arquivo no diretório de saída denominado termino.\*\*\*. Quando o módulo simbolista encontra este arquivo, ele o apaga e volta a executar (sai da malha de espera).

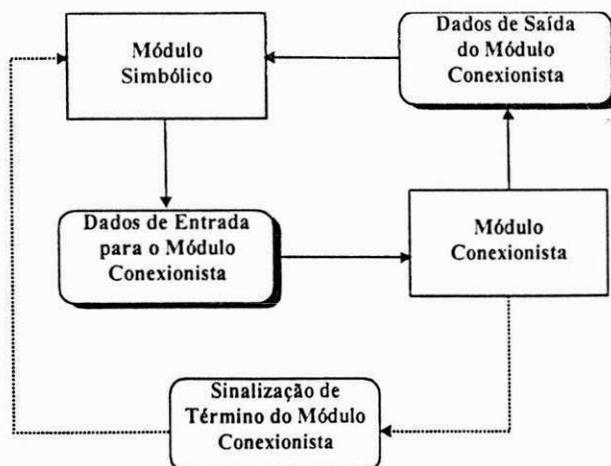


Fig. 4.2- Comunicação entre o Módulo Simbólico e o Conexionista

## 4.2 Classificação do SISNES

O SISNES, segundo a classificação de Orsier [ORS 95A], é uma abordagem híbrida com módulos separados, acoplamento estreito e subprocessamento, como é mostrado na parte em destaque da Fig.4.3. Nessa classificação a transferência de informação e controle ocorre de forma bidirecional, utiliza a memória como meio de comunicação principal e arquivos em disco para o armazenamento dos dados de comunicação.

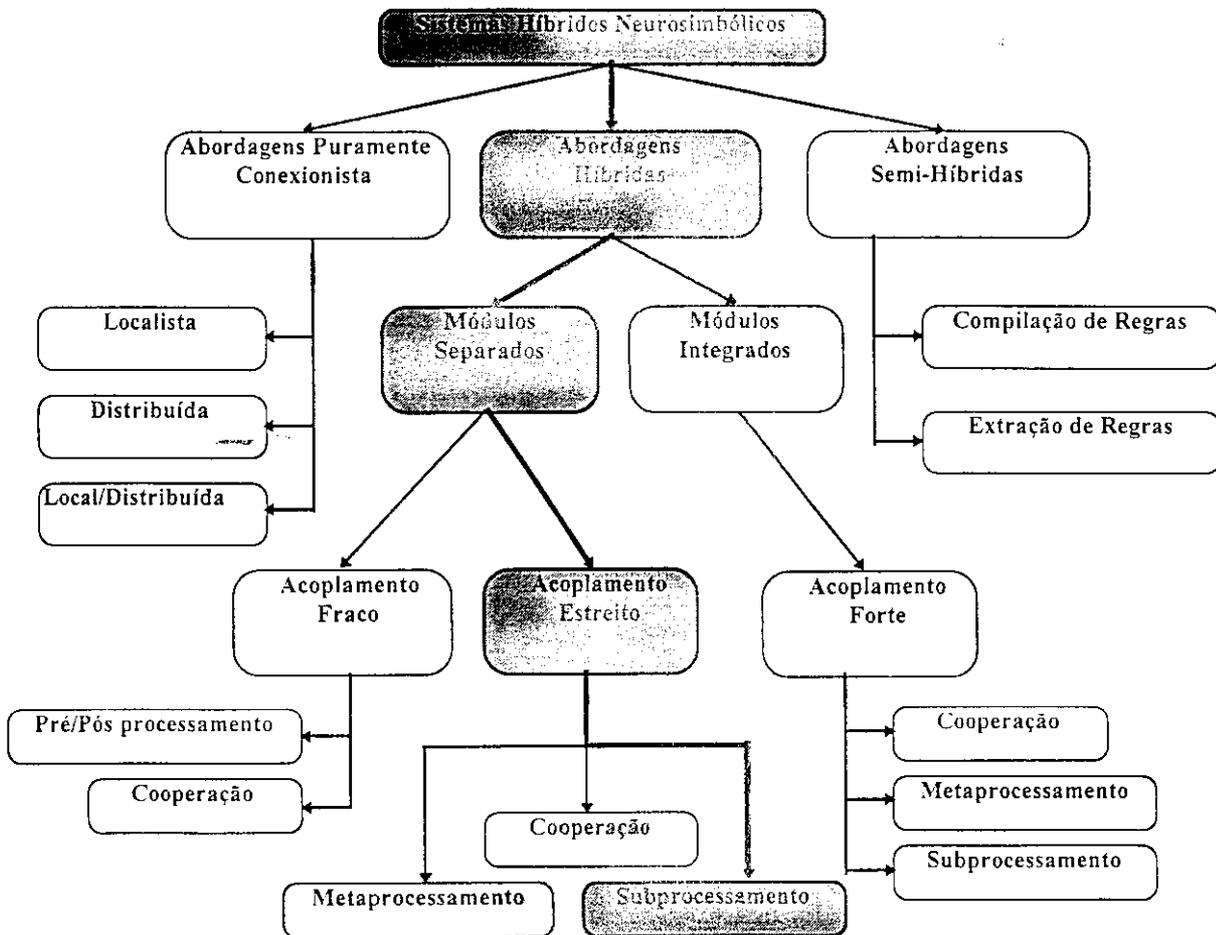


Fig. 4.3 - Localização do SISNES no diagrama de Orsier

## 4.3 Os Módulos do SISNES

O SISNES possui uma arquitetura um pouco diferente da arquitetura padrão de um sistema especialista (ver Seção 2.1.1, Fig. 2.1). Na Fig.4.4 é mostrada a arquitetura do SISNES. Nessa arquitetura todos os módulos possuem uma parte simbolista e uma parte conexionista, exceto o motor de inferência, que possui somente características simbólicas.

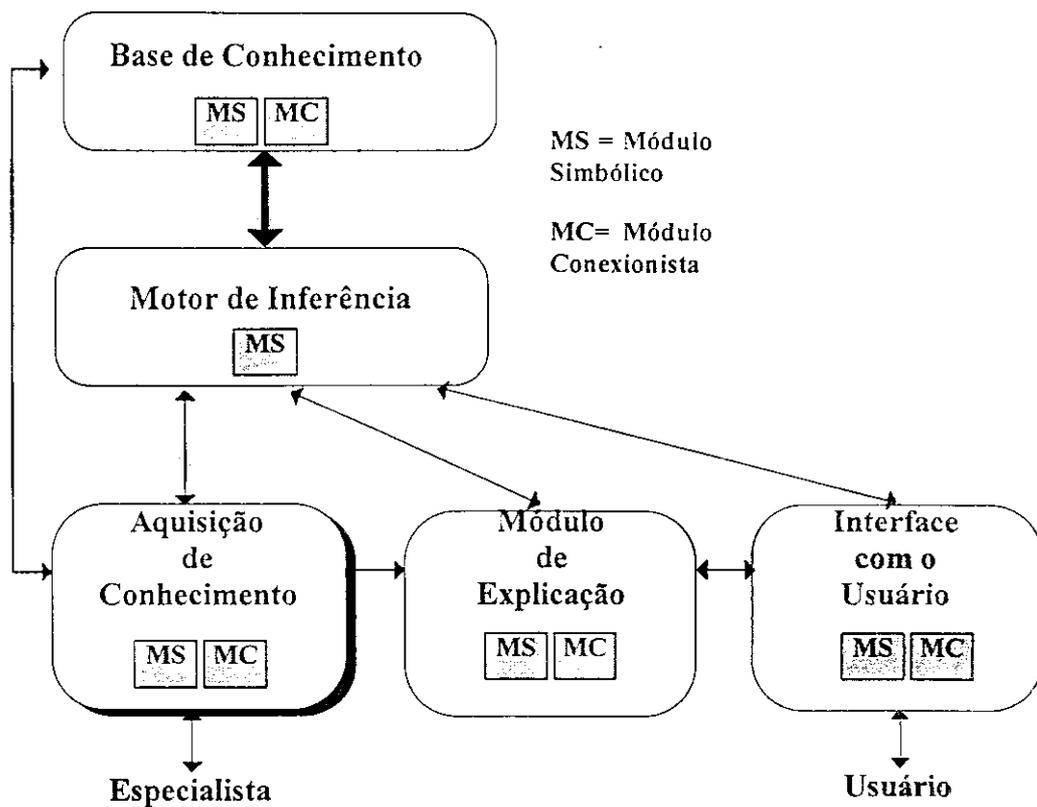


Fig. 4.4 - Módulos da Arquitetura do SISNES

### 4.3.1 Aquisição e Base de Conhecimento

A aquisição do conhecimento (Fig. 4.5) no SISNES é feita de forma independente para cada módulo, porém, o módulo simbolista deve ser desenvolvido primeiro. No módulo simbolista, a fase de aquisição do conhecimento extrai o domínio teórico do especialista, e forma a base de conhecimento. Nesta fase, ainda não existe a preocupação se as regras serão capazes de representar integralmente o problema, elas são simplesmente elaboradas (extraídas do perito). No módulo conexionista a aquisição do conhecimento é feita através de um conjunto significativo de exemplos práticos do domínio do problema. Na aquisição do conhecimento do módulo conexionista, o engenheiro do conhecimento já deve saber quais serão as regras que deverão acionar uma rede neural para resolver o problema e solicitar do especialista, que possui o domínio prático, exemplos significativos para o treinamento da rede neural.

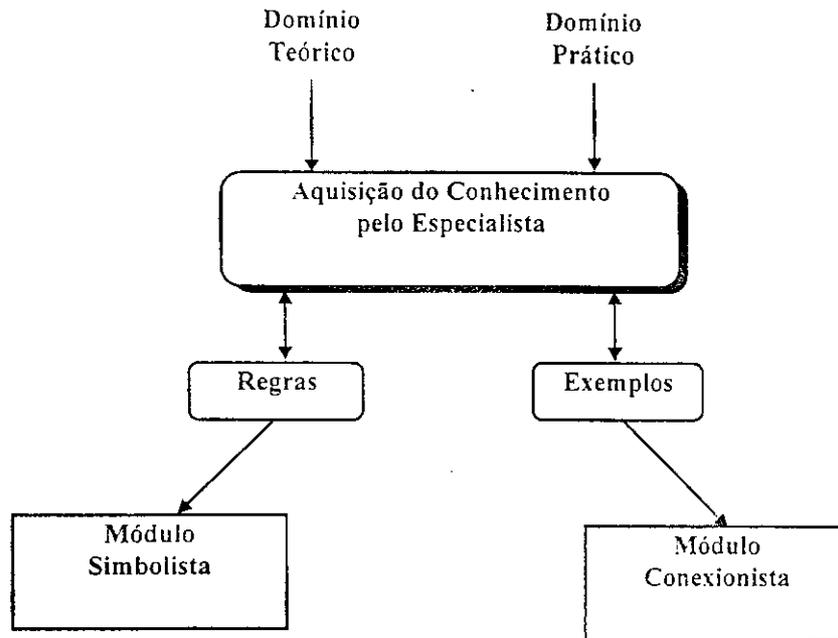


Fig. 4.5 - Aquisição do Conhecimento no SISNES

### 4.3.2 Motor de Inferência

O módulo simbólico do SISNES foi implementado utilizando o LPA-Prolog. O LPA-Prolog é uma linguagem de programação que implementa as propriedades e características da Programação em Lógica (PROLOG). A programação baseada em lógica possui regras do tipo declarativas. Num sistema inteligente o mecanismo que estabelece relacionamentos, hierarquia e manipula as regras, é chamado de motor de inferência. É através deste mecanismo que, de forma geral, se define a solução de um problema.

### 4.3.3 Módulos de Explicação

O SISNES possui explicação simbólica e conexionista. A explicação simbólica é baseada nas regras que foram utilizadas para determinar a solução final do sistema (Seção 5.2). A explicação conexionista é baseada na saída da rede neural. No SISNES o módulo conexionista é utilizado como parte da ação de uma regra e a explicação conexionista se responsabiliza por esclarecer a parte da regra para a qual o módulo conexionista foi utilizado. A explicação conexionista é feita através da execução de uma rede neural que possui a estrutura inversa da rede neural que foi utilizada para a execução da regra, chamada de falsa inversa. No Capítulo 5 será descrita a estrutura e o treinamento da falsa inversa.

### 4.3.4 Interface com o Usuário

A interação com o usuário é feita utilizando janelas, botões e marcadores no padrão windows. Inicialmente é mostrado para o usuário uma janela com perguntas, um botão para o diagnóstico e um botão para o retorno à tela anterior. O SISNES faz perguntas com relação ao problema que será tratado, ele mostra a lista de todas as perguntas e solicita que o usuário selecione marcando com um X a pergunta que deseja responder. A partir das respostas do usuário o SISNES processa o diagnóstico final. Algumas perguntas são relacionadas a uma rede neural, estas perguntas necessitam de um tipo de informação que são processadas com mais precisão através de redes neurais. A Fig. 4.6 mostra o comportamento do SISNES quando o usuário deseja fazer um diagnóstico e utiliza a seguinte representação:

1. retângulos que não estão sombreados representam as janelas,
2. retângulos sombreados e com cantos arredondados representam os botões,
3. linhas cheias apontam para os botões que as janelas possuem,
4. linhas pontilhadas mostram a ação feita se o botão for pressionado.
5. retângulos não sombreados e com cantos arredondados representa um processamento.

Quando o usuário pressionar o botão diagnóstico, o módulo simbólico entrará em ação, a solução encontrada é mostrada na janela de Resposta Final do Sistema. A janela de resposta final possui os seguintes botões:

1. **Explicação**, mostra para o usuário as regras que contribuíram para a solução através da janela de Explicação Simbolista,
2. **Voltar a Selecionar**, no caso se o usuário desejar continuar selecionando. Este botão é útil quando a resposta final indica que os dados fornecidos foram insuficiente.
3. **OK** mostra a janela de Saída com as opções de confirma a saída ou não.

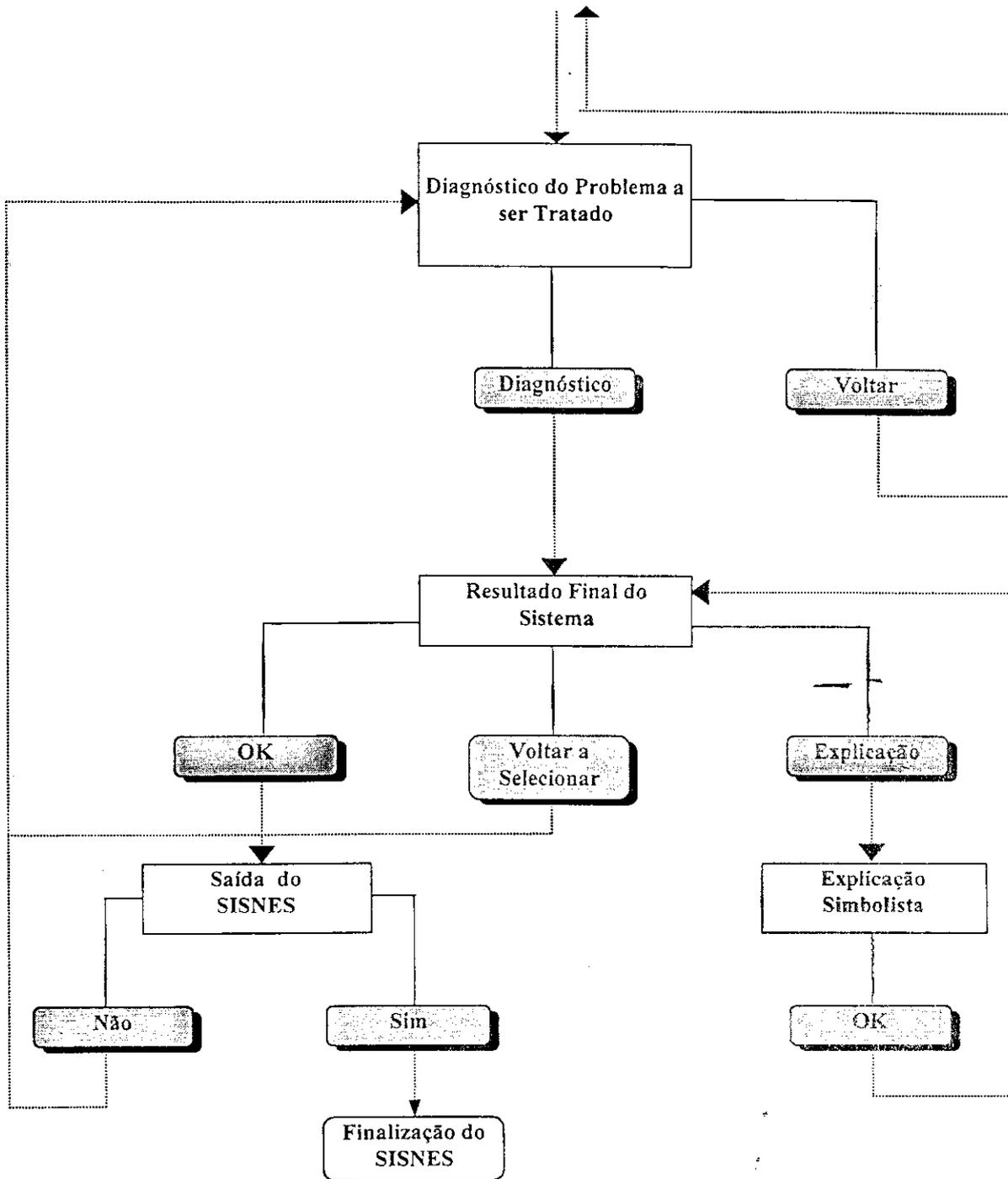


Fig. 4.6 - Comportamento do SISNES para o Diagnóstico

Quando o usuário seleciona uma pergunta que está associada a uma rede neural, o módulo conexionista pode ser ativado (Fig. 4.7), a depender da confirmação do usuário. A solução da rede neural é mostrada na janela de Resposta da Rede Neural. Após receber a resposta da rede neural, o usuário poderá selecionar os seguintes botões:

1. **Explicação**, mostra para o usuário a explicação porque a rede neural chegou ao resultado.
2. **OK** volta para a janela de diagnóstico.

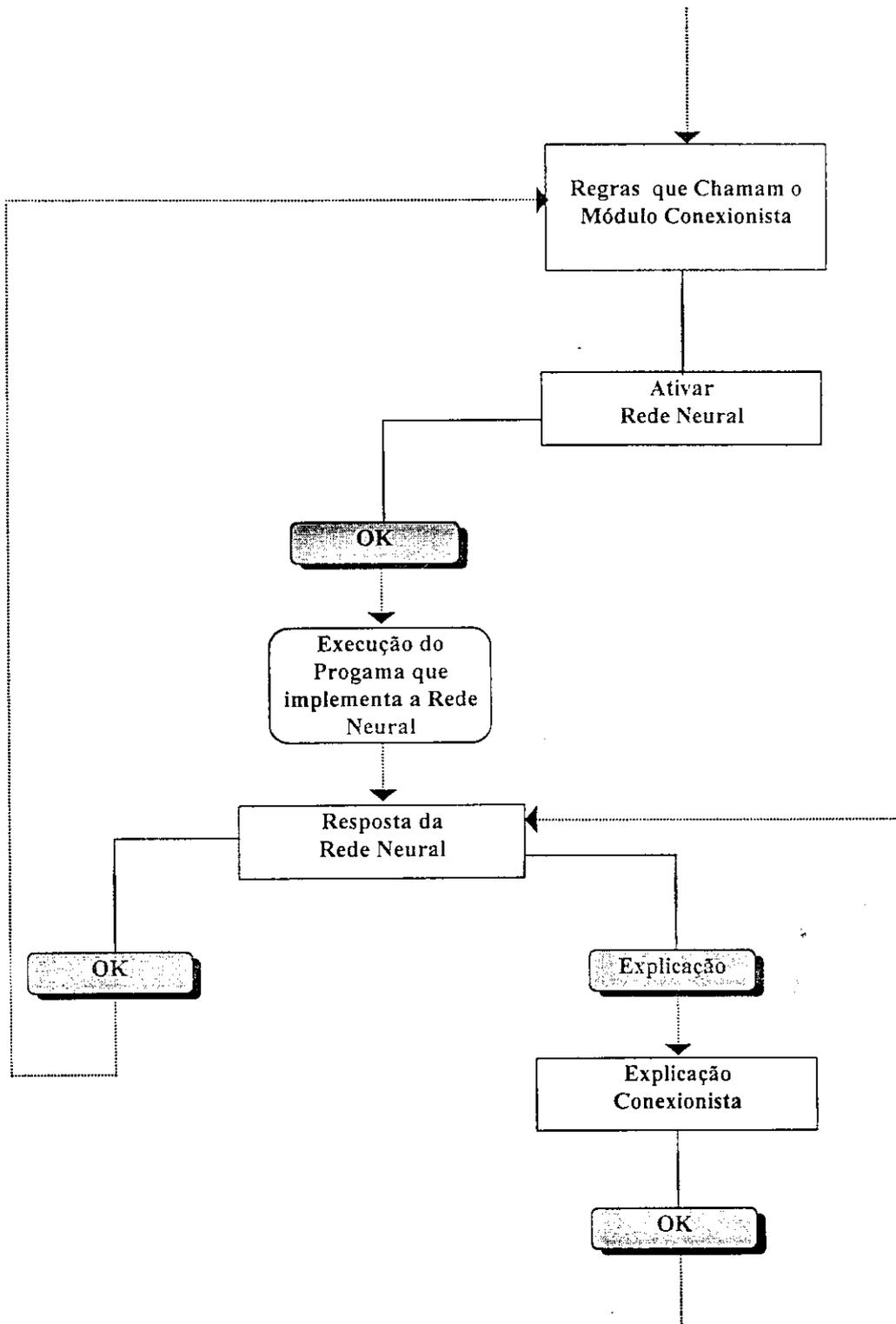


Fig. 4.7 - Comportamento do SISNES para a execução da Rede Neural

## 4.4 Implementação da Integração Neurosimbólica

Os módulos simbolista e conexionista foram implementados utilizando o LPA-prolog e o Borland C++, respectivamente. Para a implementação de todo o SISNES foram criados dois projetos: Um projeto em LPA-Prolog chamado *edificio.pl*, que contém 12 arquivos com aproximadamente 300 linhas de código. O projeto outro em Borland C++ chamado *manut.ide* com aproximadamente 1000 linhas de código. Existe a possibilidade de comunicação entre estas duas linguagens de programação. A integração neurosimbólica foi implementada da seguinte forma:

### 4.4.1 Código do LPA-Prolog

O comando Win-Prolog que possibilita a execução de um programa em outra linguagem é: *exec(Program,Args,Status)*. No SISNES o programa executável implementa uma rede neural em Borland C++, chamado de *manut.exe*. O comando EXEC no SISNES foi implementado da seguinte forma:

```
exec('diretório',S,_),
onde S é o parâmetro de entrada para o programa "manut.exe"
```

O programa *manut.exe* utiliza arquivos com as entradas da rede neural. O parâmetro S é obtido através do seguinte comando do LPA-Prolog:

```
dirbox('Selecionar Arquivo','Entrada da Rede Neural','C:\entradas\*. *',S),
```

O comando *dirbox(.)* permite que um arquivo seja selecionado e que o seu caminho e nome seja atribuído a S. O terceiro parâmetro do comando apresentado acima, é o caminho do diretório onde esta o arquivo de entrada. No SISNES todos os arquivos de entrada estão no subdiretório *entradas*.

Após a execução destes dois comandos o módulo conexionista é ativado através do programa *manut.exe*. A partir da execução do comando *exec* o programa em LPA-Prolog fica em malha de espera até a criação de um arquivo chamado de termino.\*\*\*. O código em LPA-Prolog que permite esta espera é:

```
repeat, wait(0),
dir('c:\saidas\termino.***',-32,Dir),
Dir \= [],
```

```
del('c:\saidas\termino.$$$'),
```

Quando é sinalizado o término do módulo conexionista o módulo simbolista retoma o comando e verifica qual foi a resposta do módulo conexionista através do arquivo de saída da rede neural. No SISNES o arquivo de saída do módulo conexionista é o *solucao.c* que se localiza no subdiretório *saidas*. O código que permite a execução desta tarefa é o seguinte:

```
Fin = 'c:\saidas\solucao.c',
fopen(Fin,Fin,0),
input(Fin),
get(N),
fclose(Fin),
```

#### 4.4.2 Código do C

O módulo conexionista é implementado através do programa *manut.exe*. Este programa recebe como parâmetro de entrada um arquivo que são as entradas da rede neural. O código que recebe o arquivo de entrada é:

```
strcpy(nomearq,argv[1]);
```

onde *argv[1]* é o parâmetro *S* do comando *exec* do *LPA-Prolog*

Após a execução do programa *manut.exe* o módulo conexionista deve criar um arquivo de saída, que no SISNES é chamado de *solucao.c*, e colocá-lo no subdiretório *saidas*. Esta tarefa é feita através do seguinte código:

```
fp_out = fopen("c:\\saidas\\solucao.c","w");
fprintf(fp_out,"%d \n ",sol);
onde sol é a solução da rede neural.
```

O próximo passo do módulo conexionista é sinalizar o término para o módulo simbolista. Esta sinalização é feita através da criação de um arquivo chamado *termino. \$\$\$* no subdiretório *saidas*. Em Borland C++ esta tarefa é feita da seguinte forma:

```
fp = fopen("c:\\saidas\\termino. $$$","w");
```

No próximo capítulo serão apresentados os resultados da implementação do SISNES na área de automação predial.

## Capítulo 5

### APLICAÇÃO DO SISNES

Neste capítulo será descrito uma aplicação do SISNES - SIStema NEurosimbólico usando Subprocessamento, na área de automação predial<sup>14</sup>.

#### 5.1 Automação Predial

Os sistemas de automação predial são responsáveis por controlar e supervisionar as etapas de construção do empreendimento, proporcionar serviços mais eficientes aos usuários e prover a manutenção do empreendimento. Está cada vez maior a complexidade dos empreendimentos prediais dificultando o controle manual do projeto. A utilização da Informática desde o início do projeto pode criar condições que permitam maior facilidade na implantação e instalação de serviços que proporcionam conforto aos usuários[GAR96].

Os empreendimentos que possuem sistemas de automação predial mostram consideráveis reduções econômicas no planejamento e execução do projeto e posteriormente nos gastos de manutenção tanto a nível de manutenção preventiva quanto de manutenção corretiva [JUN96].

As técnicas e definições da inteligência artificial estão sendo usadas na automação predial [GAR96]. Atualmente os “edifícios inteligentes” [JUG 96] são definidos como empreendimentos imobiliários que possuem controle de sua estrutura, funções operacionais e

manutenção, minimizando e otimizando as tarefas e custos. O SISNES foi desenvolvido e aplicado na manutenção de edificações para realizar o diagnóstico da umidade.

### 5.2 O Problema da Umidade

A umidade desafia os profissionais, pelas diversas formas de acesso às construções, pelas diversas manifestações que pode assumir e pela possibilidade de invasões múltiplas, que tornam difícil o diagnóstico de sua origem [SOU94].

A umidade pode ser entendida como o estado de úmido ou estado de molhado, é um agente deteriorante das construções. Ela está presente na natureza sob a forma sólida, líquida ou de vapor e ocorre freqüentemente nas antigas construções, sendo um grande desafio para profissionais que atuam na área de manutenção e restauro. A umidade também ocorre em edificações que não são antigas, mas que foram construídas em locais que favorecem o seu surgimento e propagação [LAC90].

Existem várias causas que podem provocar o surgimento da umidade, como por exemplo, a condensação do vapor d'água sobre as superfícies frias, a infiltração d'água através dos poros ou fissuras, a dissolução da parede causada pelas chuvas, principalmente se impelida por ventos fortes [LAC90].

O processo de umidade é caracterizado pela circulação de água, que a depender do tipo de material usado na construção, poderá causar vários danos à edificação. Se a umidade for formada por água limpa, ou seja, se não houver a presença de sais, além da erosão causada na superfície dissolvendo o material construtivo e abrindo caminho para o ataque nas camadas mais internas, poderá ocorrer a corrosão das peças metálicas. Se na água existir a presença de sais, derivado do próprio material ou do terreno, forma-se uma solução, que quando atinge a superfície se cristaliza e gera pressão deteriorando a camada externa da parede. Os sais atraem água por osmose e vapor de água por higroscopicidade, agravando problemas como a cristalização e a hidratação [LAC90].

Através do auxílio de um especialista pode-se diagnosticar o tipo de umidade e o que a causou. O diagnóstico necessita da coleta de informações sobre: o tipo de terreno onde se encontra o imóvel, o tipo de implantação, amplitude da umidade, o tipo de material da parede, localização da mancha em relação à parede, manchas aleatoriamente distribuídas, manchas

---

<sup>14</sup> Neste trabalho o termo Automação Predial corresponde principalmente a informatização das etapas de construção e de manutenção do empreendimento.

ascendentes, manchas descendentes, etc. Além dessas informações, poderão ser necessários outros tipos de informações e procedimentos como: coleta de material para análise em laboratório, medições de temperatura, medição da quantidade de sais existente no material, etc. A partir dos resultados dessas informações, o especialista poderá ser capaz de diagnosticar o tipo de umidade e saber como resolvê-la [SOU94].

Um sistema especialista que realiza o diagnóstico da umidade em edificações pode ser feito conforme um modelo simbolista codificando em regras as informações e gerando os diagnósticos[SOU 94]. Algumas informações são fáceis de serem codificadas conforme o modelo simbolista como por exemplo uma regra do tipo: **SE o teor de sais na parede está elevado Então Umidade Ascendente**[LAC 90]. Outras informações, porém, pela característica dos dados em observação e da resposta desejada, são mais difíceis e muitas vezes é impossível codificar em regras. Por exemplo, no caso de se determinar se as manifestações de umidade (manchas) crescem e de que maneira: do topo para a base da parede ou da base para o topo da parede é muito complicado e talvez a codificação em regras deste problema não consiga refletir perfeitamente a realidade.

Em casos como este, uma provável boa opção é desenvolver um sistema híbrido neurosimbólico, que utilize o método simbolista para codificar as informações que possam ser expressas através de regras e o método conexionista que trate as informações que não são codificadas em regras.

Nesta aplicação, o edifício é dividido em 5 (cinco andares), que podem ser chamados de subsolo, andar térreo, andar do meio, andar superior e cobertura. O SISNES foi utilizado para realizar o diagnóstico da umidade em edificações com base num conjunto de regras. As regras não foram obtidas diretamente de um especialista, elas foram extraídas de um sistema especialista puramente simbólico para problemas de umidade em edificações. As regras do sistema puramente simbólico foram extraídas através de entrevistas com peritos [SOU 94]. Os possíveis diagnósticos desta aplicação são: umidade ascendente, umidade descendente ou infiltração e vazamento.

A umidade ascendente ocorre principalmente nos andares térreos e no subsolo (primeiro e segundo andar), as regras que contribuem para o diagnóstico positivo de Umidade Ascendente são:

1. **SE** existe a presença de faixa horizontal de eflorêscencia E o teor de sais está elevado na base da parede.

2. **SE** existe a presença de faixa horizontal de eflorêscencia **E** o teor de sais diminui da base para o meio da parede.
3. **SE** existe a presença de faixa horizontal de eflorêscencia **E** as manchas crescem do topo para a base da parede.
4. **SE** existe a presença de faixa horizontal de eflorêscencia **E** o âmagô da parede esta úmido.
5. **SE** o âmagô da parede esta úmido **E** o teor de sais está elevado na base da parede
6. **SE** o âmagô da parede esta úmido **E** o teor de sais está elevado na base da parede **E** o teor de sais diminui da base para o meio da parede.
7. **SE** o teor de sais está elevado na base da parede **E** as manchas são uniformes em toda a parede **E** o teor de sais diminui da base para o meio da parede.
8. **SE** as manchas crescem do topo para a base da parede **E** o teor de sais está elevado na base da parede
9. **SE** as manchas crescem do topo para a base da parede **E** o teor de sais diminui da base para o meio da parede.
10. **SE** as manchas crescem do topo para a base da parede **E** o âmagô da parede está úmido

A umidade descendente ou infiltração ocorre principalmente nos andares superiores e na cobertura (quinto e quarto andar), as regras que contribuem para o diagnóstico positivo de infiltração são:

1. **SE** as manchas crescem da base para o topo da parede **E** o teor de sais está elevado no topo da parede
2. **SE** existe a presença de faixa horizontal de eflorêscencia **E** o teor de sais está elevado no topo da parede **E** o âmagô da parede está úmido
3. **SE** as manchas crescem da base para o topo da parede **E** o teor de sais está elevado no topo da parede **E** Existe mancha no topo da parede
4. **SE** as manchas crescem da base para o topo da parede **E** o teor de sais diminui do topo para o meio da parede **E** Existe mancha no topo da parede
5. **SE** as manchas crescem da base para o topo da parede **E** Existe mancha no topo da parede

Nesta aplicação com a finalidade de realizar testes, considera-se que os vazamentos ocorrem principalmente nos andares medianos do edifício (terceiro andar), a regra que contribui para o diagnóstico positivo de vazamento é:

*1. SE existe a presença de faixa horizontal de eflorêscencia E Existe um alto teor de umidade em toda a parede E o âmago da parede está úmido E as manchas são uniformes em toda a parede*

O módulo conexionista é utilizado para receber informações que são melhor processadas por redes neurais, por exemplo, determinar em que forma as manchas estão distribuídas na edificação, se é de forma crescente do topo para a base, ou de forma decrescente da base para o topo, ou não existe manchas na edificação, ou as manchas são uniformes em toda a edificação. Exemplos dessas situações são as regras de umidade ascendente 3,7,8,9 e 10.

A lógica fuzzy é utilizada como um pré-processador dos dados de entrada da regra, ela representa as informações numéricas de uma forma lingüística. Por exemplo, nas regras de umidade ascendente 1,5,6,7 e 8 a informação de entrada da regra é o valor numérico do teor de sais (entre 0 e 1), este valor é fuzzificado usando as curvas triangulares da função de pertinência (ver Seção 2.3 do Capítulo 2) que podem ser: Zero, Baixo, Médio e Elevado.

### 5.3 Codificação das Perguntas

As perguntas relativas aos andares térreos, subsolo e do meio, referentes as regras de umidade ascendente e vazamento são as seguintes:

1. Existe a presença de faixa horizontal de eflorêscencia?
2. Existe manchas na base da parede?
3. Existe alto teor de umidade em toda a parede?
4. O teor de sais está elevado na base da parede?
5. O teor de sais diminui da base para o meio da parede?
6. O âmago da parede esta úmido?
7. As manchas são uniformes em toda a parede?
8. As manchas crescem do topo para a base da parede?

As perguntas relativas aos andares superiores e cobertura, referentes as regras de infiltração são as seguintes:

1. Existe a presença de faixa horizontal de eflorêscencia?
2. Existe manchas no topo da parede?
3. Existe alto teor de umidade em toda a parede?
4. O teor de sais está elevado no topo da parede?
5. O teor de sais diminui do topo para o meio da parede?
6. O âmago da parede esta úmido?
7. As manchas são uniformes em toda a parede?
8. As manchas crescem da base para o topo da parede?

Nesta dissertação só serão apresentadas as implementações das regras e das perguntas para o diagnóstico de umidade ascendente no subsolo, nos outros andares a implementação é similar. As perguntas  $S_n$  ( $1 \leq n \leq 8$ ) relativas ao subsolo foram implementadas da seguinte forma:

S1 = Existe a presença de faixa horizontal de eflorêscencia?

S2 = Existe manchas na base da parede? — —

S3 = Existe alto teor de umidade em toda a parede?

S4 = O teor de sais está elevado na base da parede (fuzzy)?

S5 = O teor de sais diminui da base para o meio da parede?

S6 = O âmago da parede esta úmido?

S7 = As manchas são uniformes em toda a parede (Rede Neural)?

S8 = As manchas crescem da base para o topo da parede (Rede Neural)?

O  $S_n$  recebe o valor 1 quando a pergunta for seleccionada (marcada com um X).

Na seção 5.6 serão mostrados exemplos da execução do módulo conexionista referente as perguntas S7 e S8. Na seção 5.7 será mostrado um exemplo da utilização da lógica fuzzy referente a pergunta S4.

## 5.4 Codificação das Regras

As regras do módulo simbolista foram implementadas através de cláusulas no Win-Prolog. As variáveis lógicas  $R_i$  das regras recebem o valor 1 quando as perguntas a elas relacionadas também possuem o valor 1, caso contrário as regras recebem o valor 0. As regras listadas abaixo são as mostradas no diagnóstico de umidade ascendente (Seção 5.2) na forma codificada.

```
regra1(S1,S4,R1) :-  
    S1 == 1,S4 == 1,  
    R1 = 1,  
    !.
```

```
regra1(S1,S4,R1) :-  
    R1 = 0,  
    !.
```

```
regra2(S1,S5,R2) :-  
    S1 == 1,S5 == 1,  
    R2 = 1,  
    !.
```

```
regra2(S1,S5,R2) :-  
    R2 = 0,  
    !.
```

```
regra3(S1,S8,R3) :-  
    S1 == 1,S8 == 1,  
    R3 = 1,  
    !.
```

```
regra3(S1,S8,R3) :-  
    R3 = 0,  
    !.
```

```
regra4(S1,S6,R4) :-  
    S1 == 1,S6 == 1,  
    R4 = 1,  
    !.
```

```
regra4(S1,S6,R4) :-  
    R4 = 0,  
    !.
```

```
regra5(S4,S6,R5) :-  
    S4 == 1,S6 == 1,  
    R5 = 1,  
    !.
```

```
regra5(S4,S6,R5) :-  
    R5 = 0,  
    !.
```

```
regra6(S4,S5,S6,R6) :-  
    S4 == 1,S5 == 1,S6 == 1,  
    R6 = 1,  
    !.
```

```
regra6(S4,S5,S6,R6) :-  
    R6 = 0,  
    !.
```

```
regra7(S4,S5,S7,R7) :-  
    S4 == 1,S5 == 1,S7 == 1,  
    R7 = 1,  
    !.
```

```
regra7(S4,S5,S7,R7) :-  
    R7 = 0,  
    !.
```

```
regra8(S4,S8,R8) :-  
    S4 == 1,S8 == 1,  
    R8 = 1,  
    !.
```

```
regra8(S4,S8,R8) :-  
    R8 = 0,  
    !.
```

```
regra9(S5,S8,R9) :-  
    S5 == 1,S8 == 1,  
    R9 = 1,  
    !.
```

```
regra9(S5,S8,R9) :-  
    R9 = 0,  
    !.
```

```
regra10(S6,S8,R10) :-  
    S6 == 1,S8 == 1,  
    R10 = 1,  
    !.
```

```
regra10(S6,S8,R10) :-  
    R10 = 0,  
    !.
```

A cláusula diagnóstico usa os valores binários de Ri gerados pelas regras do módulo simbolista. O diagnóstico do SISNES informará o tipo da umidade. As cláusulas para o diagnóstico de Umidade Ascendente são as seguintes:

```
diagnostico(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,RR) :-  
    R1 == 1,  
    RR = ` Umidade Ascendente.`,  
    !.
```

```
diagnostico(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,RR) :-  
    R2 == 1,  
    RR = ` Umidade Ascendente.`,  
    !.
```

```
diagnostico(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,RR) :-  
    R3 == 1,  
    RR = ` Umidade Ascendente.`,  
    !.
```

```
diagnostico(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,RR) :-  
    R4 == 1,  
    RR = ` Umidade Ascendente.`,  
    !.
```

```
diagnostico(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,RR) :-  
    R5 == 1,  
    RR = ` Umidade Ascendente.`,  
    !.
```

```
diagnostico(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,RR) :-  
    R6 == 1,  
    RR = ` Umidade Ascendente.`,  
    !.
```

```
diagnostico(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,RR) :-  
    R7 == 1,  
    RR = ` Umidade Ascendente.`,  
    !.
```

```
diagnostico(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,RR) :-  
    R8 == 1,  
    RR = ` Umidade Ascendente.`,  
    !.
```

```
diagnostico(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,RR) :-  
    R9 == 1,  
    RR = ` Umidade Ascendente.`,  
    !.
```

```
diagnostico(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,RR) :-  
    R10 == 1,  
    RR = ` Umidade Ascendente.`,  
    !.
```

Os diagnósticos sobre a umidade descendente (infiltração) e sobre os vazamentos não serão mostrados pois são similares ao diagnóstico de umidade ascendente.

## 5.5 Inicialização do SISNES

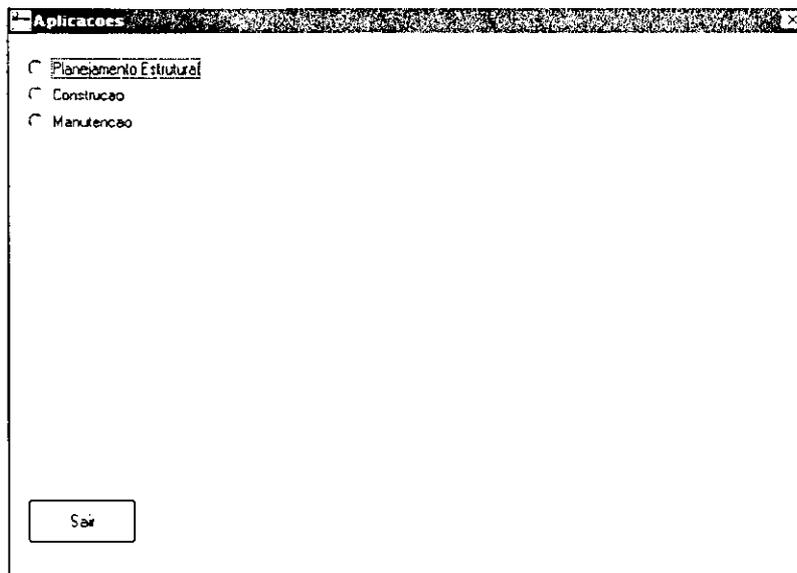
O SISNES é executado pelo programa *edificio.pl* do LPA Win-Prolog. A tela inicial do SISNES é mostrada na Fig. 5.1.



Fig. 5.1 - Tela Inicial do SISNES

Após o usuário selecionar a tecla 'OK', o SISNES apresentará a tela das aplicações que são: Planejamento Estrutural, Construção e Manutenção (Fig. 5.2).

O Planejamento Estrutural é a fase inicial do projeto de uma edificação. Nesta fase considera-se que são especificados e planejados os sistemas informatizados que serão usados. A fase Construção se refere à fase da construção da edificação. A manutenção é a fase em que a edificação já está construída e necessita de manutenção nas redes elétrica e hidro-sanitária, e na alvenaria. Nesta dissertação só é apresentado a aplicação Manutenção. O usuário pode finalizar a consulta ao SISNES teclando o botão 'Sair'.



**Fig. 5.2 - Tela de Aplicações do SISNES**

Após o usuário selecionar a opção Manutenção, o SISNES apresentará a tela "Manutenção" (ver Fig. 5.3) que são: Rede Elétrica, Rede Hidro-sanitária e Alvenarias. Neste trabalho foi implementado o módulo que se refere à manutenção das alvenarias mais especificamente no diagnóstico da umidade na edificação. O usuário pode retornar à tela de aplicações do SISNES teclando o botão "Voltar".

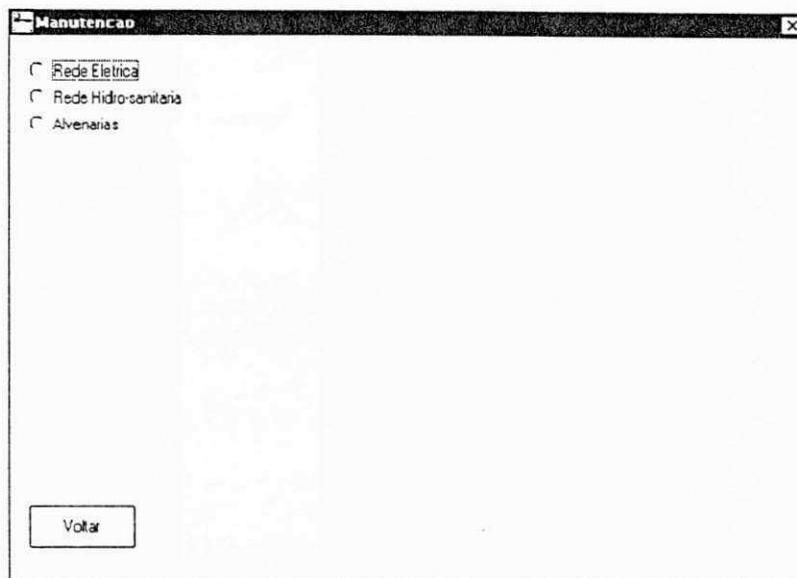


Fig. 5.3 - Tela de Manutenção do SISNES

## 5.6 Análise da Umidade do Edifício

Após o usuário escolher a opção de Alvenarias, o SISNES apresentará a tela Diagnóstico da Umidade do Edifício (ver Fig. 5.4). Nesta tela os usuários recebem as informações gerais do edifício e informam como estão distribuídas as manchas causadas pela umidade nos andares do edifício. Além disso, o usuário poderá escolher a opção “Voltar” para retornar à tela de Manutenção do SISNES. Após o usuário fornecer todas as informações sobre a umidade nas paredes do edifício, ele poderá pedir ao SISNES o diagnóstico final teclando o botão “Diagnóstico”.

O usuário tem duas formas para especificar as manchas nos andares do edifício. Na primeira forma, o usuário especifica diretamente a umidade existente em cada andar (as cinco primeiras opções). A segunda forma, representada por “Distribuição das manchas no Edifício (Rede Neural)”, permite que o usuário use uma rede neural com os arquivos de dados previamente preparados para o diagnóstico.

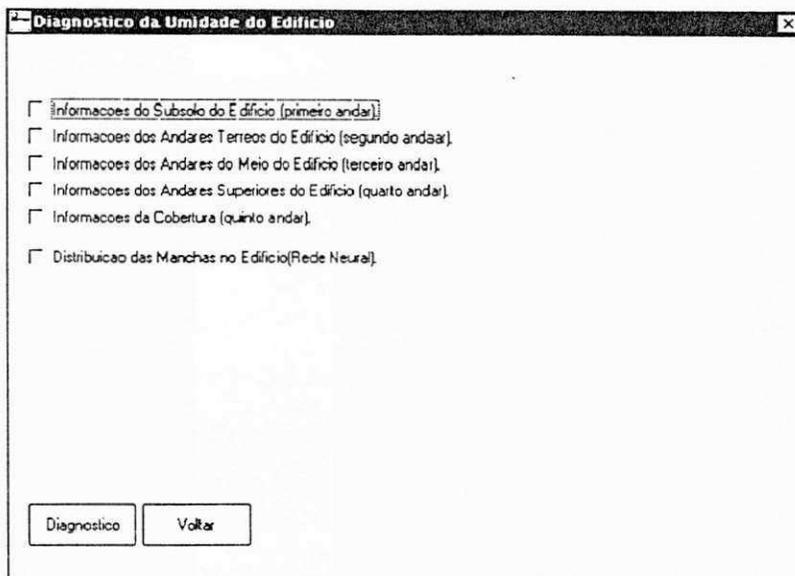


Fig 5.4 - Tela do Diagnóstico de Umidade em Edificações

### 5.6.1 Informações sobre os Andares do Edificio

Por exemplo, se o usuário escolher a opção “Informações do Subsolo do Edificio (primeiro andar)”, o SISNES apresentará a tela “Informações do Subsolo do Edificio (primeiro andar)” que se refere às perguntas relativas ao subsolo (Fig.5.5). O usuário poderá selecionar as perguntas 1, 5 e 6 marcadas na Fig. 5.5.

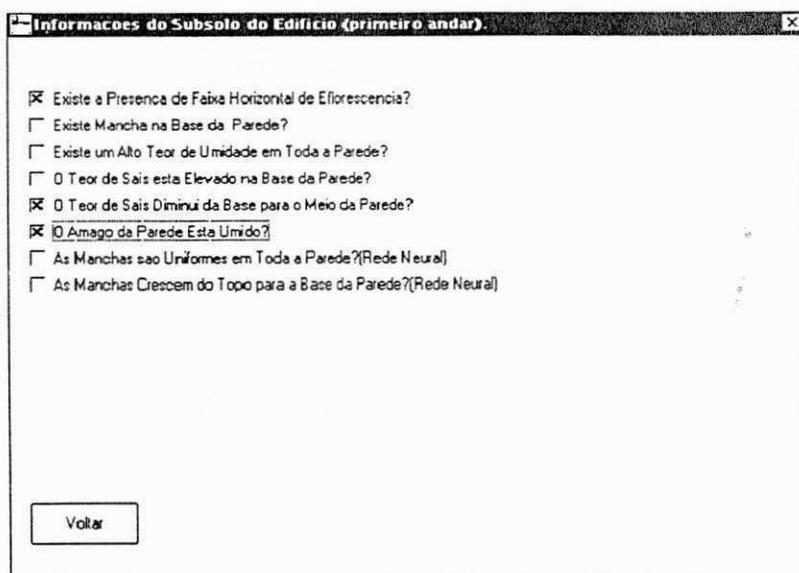


Fig. 5.5 - Tela das Perguntas para o Diagnóstico no Subsolo

A seguir o usuário deverá teclar a opção “Voltar”, retornando à tela de “Diagnóstico de Umidade Ascendente” mostrada na Fig 5.4.

### 5.6.2 Diagnóstico da Umidade no Edifício

Se o usuário estiver satisfeito e não tiver mais informações sobre a umidade do edifício, ele poderá teclar a opção “Diagnóstico” na tela “Diagnóstico de Umidade Ascendente” (Fig. 5.4). A seguir o SISNES apresenta na mesma tela da Fig. 5.4, a tela “Resposta Final do Sistema”, onde é apresentado o diagnóstico que neste caso é positivo (Umidade Ascendente, Fig. 5.6). O SISNES utilizou as regras 2 e 4 para esse diagnóstico.

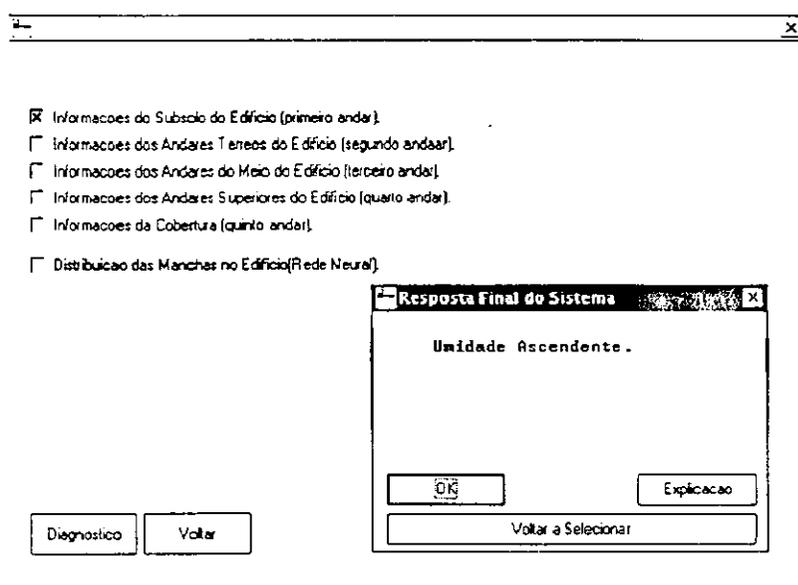


Fig. 5.6 - Tela do Resultado Final do SISNES

Após o diagnóstico apresentado pelo SISNES, o usuário terá três opções. A primeira é se ele não estiver satisfeito com o diagnóstico e desejar continuar a fornecer informações, ele teclará o botão “Voltar a Selecionar” da tela “Resposta Final do Sistema”. A segunda opção é quando o usuário está satisfeito com o diagnóstico e deseja finalizar a consulta, ele faz isso teclando o botão “OK” da tela “Resposta Final do Sistema”. A terceira opção é quando o usuário deseja uma explicação do SISNES para o diagnóstico. Nesse caso, ele solicita a explicação teclando o botão “Explicação” da tela “Resposta Final do Sistema”. Por exemplo, se o usuário escolher a opção Explicação, o SISNES apresenta a tela da “Explicação Simbólica” do diagnóstico apresentada na Fig. 5.7. A explicação simbólica irá informar ao usuário quais foram as regras utilizadas para o diagnóstico final.

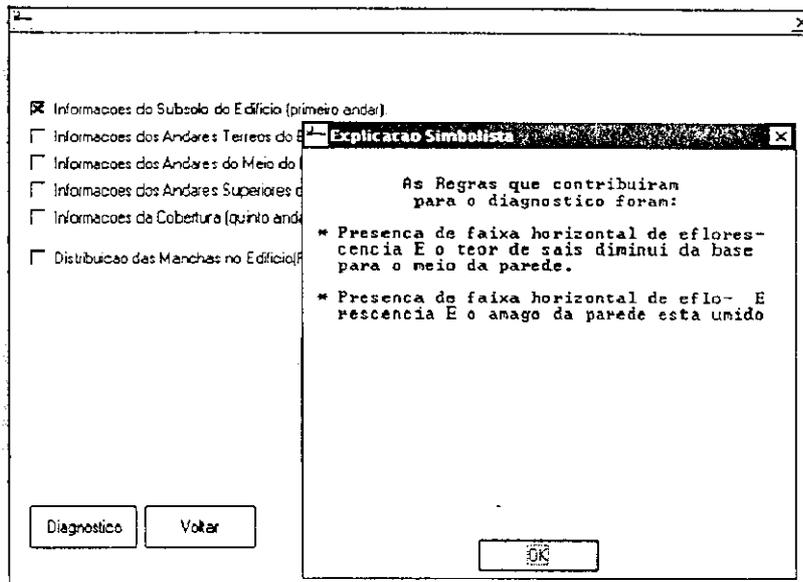


Fig. 5.7 - Tela da Explicação Simbólica do SISNES

### 5.6.3 Execução do Módulo Conexcionista

Uma situação onde será executado o módulo conexcionista é quando o usuário na tela de “Informações do Subsolo do Edifício” (Fig. 5.5), seleciona uma pergunta que requer informações da forma como as manchas estão distribuídas. Esta informação é processada com maior eficiência através de uma rede neural.

Um exemplo é apresentado na Fig. 5.8 onde a pergunta 8, “Manchas Crescem do Topo para a Base da Parede? (Rede Neural)”, foi selecionada. Após a seleção, a tela “Módulo Conexcionista” é sobreposta à tela “Informações do Subsolo do Edifício”. O usuário deverá teclar o botão “OK” da tela “Módulo Conexcionista” para o início da execução do módulo conexcionista.

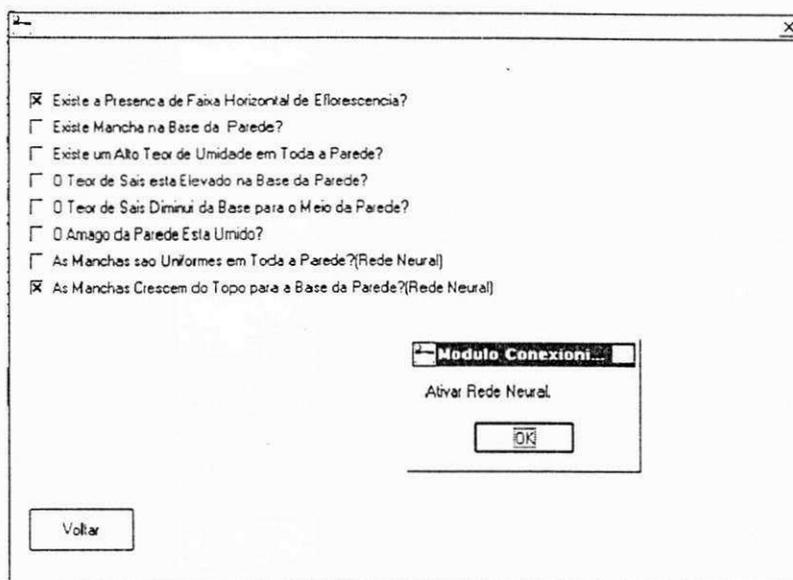


Fig. 5.8 - Tela para Execução do Módulo Conexioni

Antes da execução do módulo conexioni, o usuário deve preparar um arquivo texto no diretório *entradas* com os cinco valores do tamanho das manchas, normalizados entre 0 e 1. Os valores do arquivo serão usados como entradas para a rede neural, esta forma de entradas de dados foi utilizada para validação do SISNES, seria mais interessante a entrada de dados diretamente pelo usuário durante a execução do sistema.

Após o usuário teclar “OK”, é apresentada a tela (Fig. 5.9) “Selecionar Arquivo - (Entrada da Rede Neural)”. O usuário deverá localizar e selecionar o arquivo de entrada da rede neural e teclar o botão “OK”.

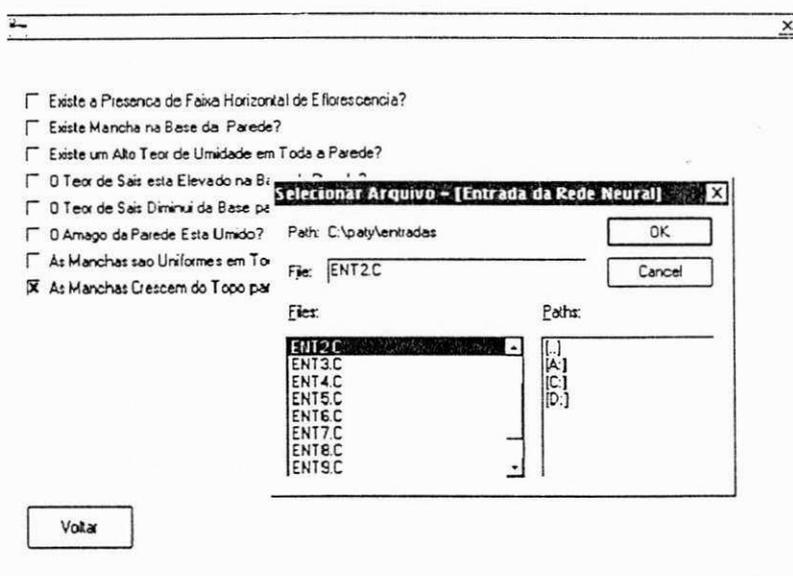


Fig. 5.9 - Tela para Seleção da Entrada de Rede Neural

A execução do módulo conexcionista é feita através da rede neural mostrada na Fig. 5.10. A rede neural do módulo conexcionista foi implementada (em Borland C++) formando uma rede neural multicamadas com 3 camadas: 5 neurônios na camada de entrada, 10 neurônios na camada oculta e 2 neurônios na camada de saída. A estrutura da rede foi obtida através de testes e tentativas na fase de treinamento. Os neurônios da camada de entrada implementam funções lineares, os neurônios da camada oculta implementam funções sigmoide e os da camada de saída funções tangente hiperbólicas.

Para a comunicação entre os módulos simbolistas e conexcionistas, os valores das entradas da rede neural, preparadas no módulo simbolista, são armazenadas num arquivo do diretório *entradas*(Fig. 5.9).Todas as saídas da rede neural são colocadas, pelo módulo conexcionista, num arquivo do diretório chamado *saídas*.

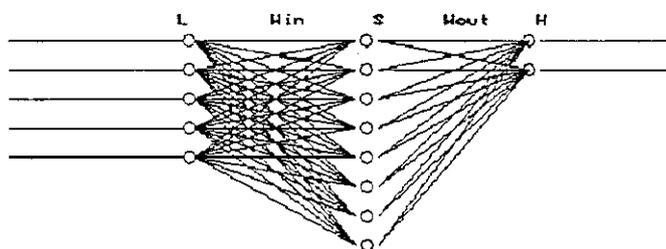


Fig. 5.10 - Rede Neural do Módulo Conexcionista

Na fase de treinamento da rede neural foram utilizados 10 exemplos de manchas distribuídas numa edificação. O tipo do treinamento realizado foi o supervisionado; as saídas da rede neural foram comparadas com a saída objetivo. Foram necessárias 1.000 (mil) interações para a rede neural aprender satisfatoriamente os exemplos.

Os dados do treinamento são mostradas na Fig. 5.11 dispostos em colunas. Cada coluna possui três campos: o campo *Saídas da Rede* que mostra os valores calculados pela rede neural, o campo *Objetivo* que representa os valores que a rede neural deve calcular na sua saída. No terceiro campo são mostrados os valores das entradas da rede neural, conjunto de treinamento. Esses valores representam o tipo de manchas encontradas nas paredes do edifício.

SAIDAS DA REDE										
OBJETIVO										
ENTRADAS										

Fig. 5.11 Conjunto de Treinamento da Rede Neural

Na Fig. 5.11 as manchas das colunas 1 a 4 estão em ordem crescente da base para o teto da parede, as manchas das colunas 5 a 8 estão em ordem decrescente da base para o teto da parede. As manchas das colunas 4 e 8 são exemplos de distribuição em que não é muito visível a direção do crescimento das manchas. Na coluna 9 são mostrados os resultados obtidos quando não existe mancha na parede. Na coluna 10 são mostrados os resultados obtidos quando existe mancha em toda a parede. As respostas da rede neural estão muito próximas das respostas objetivo. Pode-se observar que a rede neural aprendeu satisfatoriamente os exemplos fornecidos.

O módulo simbolista aciona o módulo conexionista com as declarações da linguagem PROLOG descritas na Seção 4.4.1 do Capítulo 4. Supondo-se que os dados do arquivo de entrada referentes ao tamanho das manchas são: 0.9, 0.7, 0.5, 0.3 e 0.1, a tela do resultado fornecido pela rede neural do módulo conexionista é mostrado na Fig. 5.12.

Dados de Entrada: 0.90 0.70 0.50 0.30 0.10

Saída da rede Neural: 0.9799 -0.2304

Resultado: Manchas Crescentes do Topo para a Base da Parede

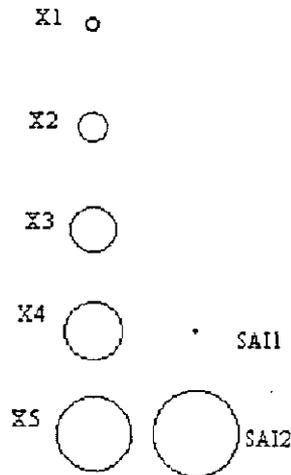


Fig. 5.12 - Resultado da Rede Neural

Na Fig. 5.12 são apresentados o conjunto dos valores numéricos das entradas (cinco) e das saídas (duas) da rede neural. Esses conjuntos de entrada e de saída são representados na forma de círculos. Os diâmetros dos círculos são proporcionais aos valores de entrada ou de saída. Os dois neurônios da camada de saída da rede neural são do tipo Tangente Hiperbólico, onde as suas saídas variam entre -1 e 1. O módulo connexionista transforma as saídas da rede neural em valores lógicos "0" e "1", isto é  $SAI_1 = "0"$  e  $SAI_2 = "1"$ , e os envia ao módulo simbolista.

A Tab. 5.1, usada pelos módulos connexionista e simbolista, relaciona as saídas da rede neural com o tipo de manchas.

$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$SAI_1$	$SAI_2$	Resultado
0.1	0.3	0.5	0.7	0.9	"0"	"1"	MCTBP
0.9	0.7	0.5	0.3	0.1	"1"	"0"	MCBTP
0.	0.	0.	0.	0.	"0"	"0"	NHMP
0.9	0.9	0.9	0.9	0.9	"1"	"1"	MUTP

Tab. 5.1 - Resultados da Rede Neural do Módulo Conexionista

Os resultados possíveis mostrados na Tab. 5.1 são:

- MCTBP → Manchas Crescentes do Topo para a Base da Parede
- MCBTP → Manchas Crescentes da base para o Topo da Parede
- NHMP → Não Há Manchas na Parede
- MUTP → Manchas Uniformes em Toda a Parede

Após a finalização do módulo conexcionista, o módulo simbolista recebe os valores binários das duas saídas da forma descrita na Seção 4.4.1. o SISNES apresenta o resultado desse módulo na tela “Resposta da Rede Neural” (Fig. 5.13). Através da Tab.5.1, o módulo simbolista informa ao usuário que as manchas são crescentes do topo para a base da parede (Fig. 5.13).

Após o resultado conexcionista apresentado pelo SISNES (Fig. 5.13), o usuário terá duas opções. Na primeira opção, se ele estiver satisfeito com a resposta conexcionista e deseja continuar a consulta, ele tecla o botão “OK” da tela “Resposta da Rede Neural” e retorna a tela “Informações do Subsolo do Edifício (Fig. 5.5). Na segunda opção, o usuário solicitará a explicação do resultado ao módulo conexcionista, teclando o botão “Explicação” da tela “Resposta da Rede Neural” .

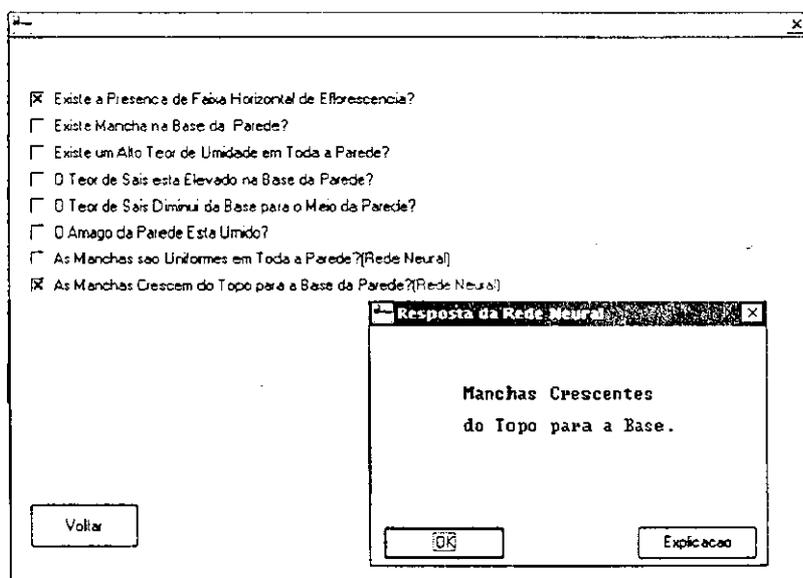


Fig. 5.13 - Tela do Resultado Conexcionista

### 5.6.3.1 Explicação Conexionista

A explicação conexionista apresentada pelo SISNES é feita através do módulo conexionista utilizando a falsa inversa da rede neural. A falsa inversa da rede neural foi implementada em uma rede neural multicamadas com 3 camadas: 2 neurônios na camada de entrada, 10 neurônios na camada oculta e 5 neurônios na camada de saída. A estrutura da falsa inversa foi obtida através da inversão das entradas e das saídas da rede neural vista na Seção 5.6.3. Os neurônios da camada de entrada implementam funções lineares, os neurônios da camada oculta implementam funções sigmoide e o da camada de saída funções tangente hiperbólicas.

Os valores das entradas da falsa inversa da rede neural são os valores binários fornecidos pela saída da rede neural utilizada no resultado do módulo conexionista (Fig. 5.14).

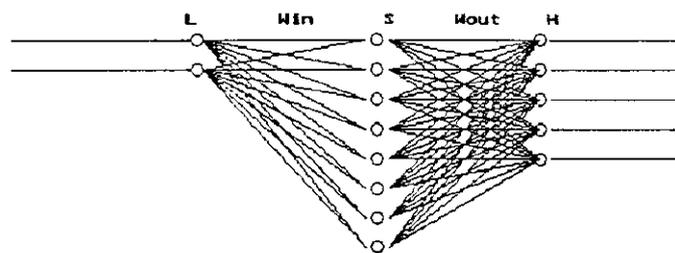


Fig. 5.14 - Inversa da Rede Neural

Na fase de treinamento da falsa inversa foram utilizados os 10 exemplos das manchas que também foram utilizados no treinamento da rede neural vista na Seção 5.6.3, onde as entradas da falsa inversa correspondem as saídas da rede neural e as saídas da rede neural são as entradas da falsa inversa (Fig. 5.15). O tipo do treinamento realizado foi o supervisionado. Foram necessárias 1.000 (mil) interações para a falsa inversa aprender satisfatoriamente os exemplos.

ENTRADAS	
SAIDAS DA REDE	

Fig. 5.15 Conjunto de Treinamento da Falsa Inversa

A saída da falsa inversa são os valores numéricos das manchas na parede, esses valores após a execução da rede neural são apresentados na tela do resultado da explicação conexionista (Fig. 5.16).

```
Dados de Entrada: 1.0000  0.0000
Saída da rede Neural: 0.8757  0.7274  0.5262  0.3098  0.1159
Resultado: Manchas Crescentes do Topo para a Base da Parede
```

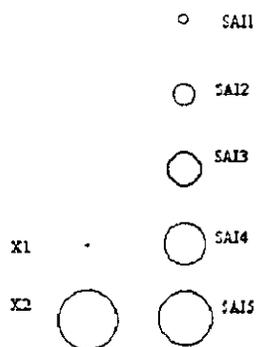


Fig. 5.16 - Explicação do Resultado da Rede Neural

Na Fig. 5.16, as saídas da rede neural apresentam os padrões (numéricos e em círculos) das manchas correspondente às duas entradas binárias.

O módulo conexionista envia os valores das cinco saídas para o módulo simbolista. O SISNES apresenta o resultado do módulo conexionista na tela “Explicação Conexionista” (Fig. 5.17). O usuário deverá teclar o botão “OK” da tela “Explicação Conexionista” para continuar a consulta no SISNES.

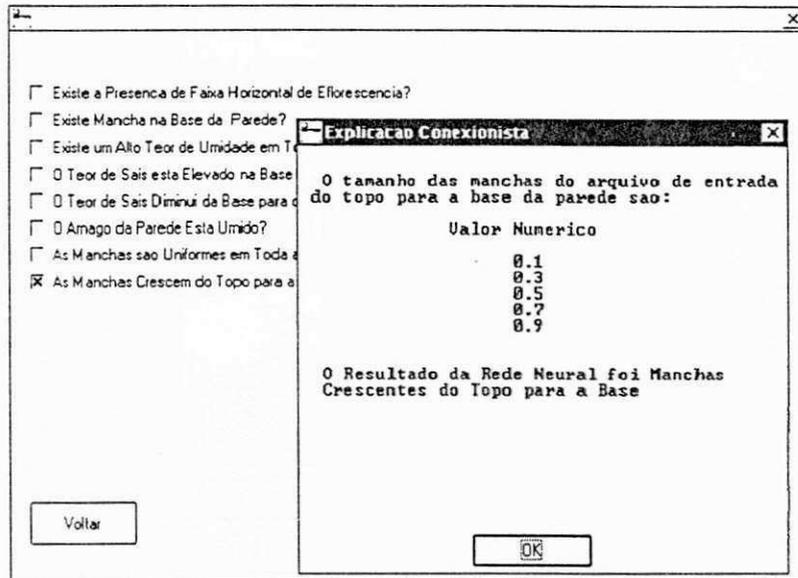


Fig. 5.17 - Tela da Explicação Conexionista

#### 5.6.4 - Utilização da Lógica fuzzy

Uma situação que utiliza a lógica fuzzy é quando o usuário seleciona uma pergunta que requer informações sobre o teor de sais na parede. Um exemplo é apresentado na Fig. 5.18 onde a pergunta 4, “O Teor de Sais está Elevado na Base da Parede?”, foi selecionada. O SISNES apresenta, sobreposta à tela da Fig. 5.5, a tela “Fuzzy” que solicita dados de entrada para a utilização da lógica fuzzy. Nessa tela, o usuário deverá selecionar o valor numérico entre 0 e 1 do teor de sais (TS) e teclar o botão “OK”. O resultado da lógica fuzzy é o valor lingüístico (TSf) do valor numérico do teor de sais.

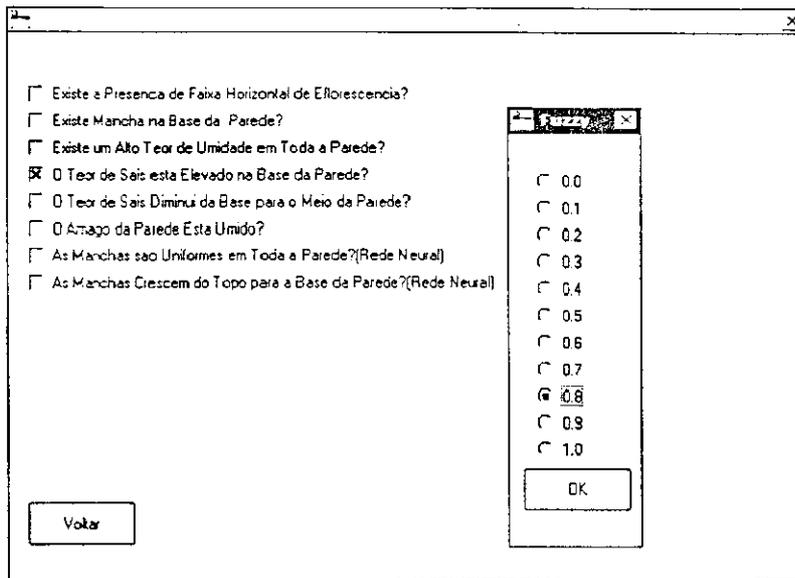
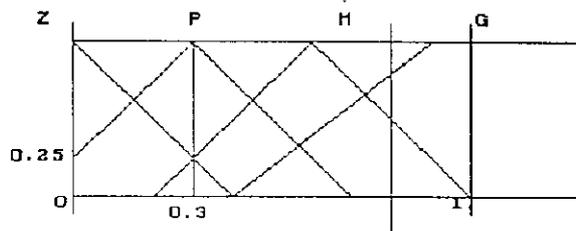


Fig. 5.18 - Tela de Solicitação da Entrada Fuzzy

O valor lingüístico (TSf) é obtido através da avaliação das funções de pertinência do TS (ver Fig. 5.19). Na Fig. 5.19 a abscissa é o valor numérico de TS normalizado entre 0 e 1. A ordenada representa valores das funções de pertinência (Z, P, M e G) referentes ao TSf. As funções de pertinência de TSf são triangulares. Elas são representadas por retas descritas pela equação  $Y = Ax + B$ , onde  $A$  representa a sua declividade e  $B$  o seu deslocamento.

Na Fig. 5.18 o valor 0.8 foi selecionado. Após o usuário teclar o botão "OK", o módulo connexionista é acionado e a tela da Fig. 5.19 é apresentada.

Na Fig. 5.19 são mostradas as curvas triangulares referentes às funções de pertinência de TSf. Por exemplo, para o cálculo da função de pertinência P, foram usadas as equações:  $Yp = Ax + Bp$  e  $Yn = -Ax + Bn$ . A declividade das retas de P é dada por  $A(2) = \pm 2,5$ . A primeira reta de P (da esquerda para a direita) tem a declividade positiva e a segunda tem a declividade negativa. O valor de TS, na abscissa da curva da Fig. 5.19, que faz  $Yp = 1$  e  $Yn = 1$ , é  $X(2) = 0,3$



$Y = Ax + B$		$U_{\text{vencedor}} = G \quad G=1$
$X(1) = 0.00 \quad A(1) = 2.50$		$Memberships(Z) = 0.00$
$X(2) = 0.30 \quad A(2) = 2.50$		$Memberships(P) = 0.00$
$X(3) = 0.60 \quad A(3) = 2.50$		$Memberships(H) = 0.50$
$X(4) = 0.90 \quad A(4) = 2.00$		$Memberships(G) = 0.80$

Fig. 5.19 - Função de Pertinência Fuzzy

Para o cálculo dos parâmetros  $B_p$  e  $B_n$  das duas equações das retas de P é feito pelo módulo conexcionista da seguinte forma:

- Reta com declividade positiva  $Y_p = Ax + B_p$ , para  $Y_p = 1$  e  $A = 2,5$ , obteve-se  $B_p = 0,25$ . Então  $Y_p = 2,5x + 0,25$ .
- Reta com declividade negativa  $Y_n = -Ax + B_n$ , para  $Y_n = 1$  e  $A = 2,5$ , obteve-se  $B_n = 1,75$ . Então  $Y_n = -2,5x + 1,75$ .

Os valores de  $Y_p$  e  $Y_n$  devem ser positivos e menor ou igual a 1. As outras funções de pertinência de TSf foram calculadas de forma similar à função P. No exemplo da Fig. 5.19, com  $TS = 0,8$ , obtiveram-se os valores das funções de pertinência mostradas à direita da Fig. 5.19. O maior valor dessas funções foi o  $Memberships(G) = 0.8$  que representa  $TSf = G$ .

O resultado da lógica fuzzy foi G (Grande), para o cálculo do teor de sais. No módulo simbolista a função de pertinência fuzzy G, corresponde a *Elevado*. A Fig. 5.20 apresenta a tela do resultado da lógica fuzzy no módulo simbolista. O usuário deverá teclar o botão "OK" da tela "Valor Lingüístico do Teor de Sais" para continuar a consulta no SISNES.

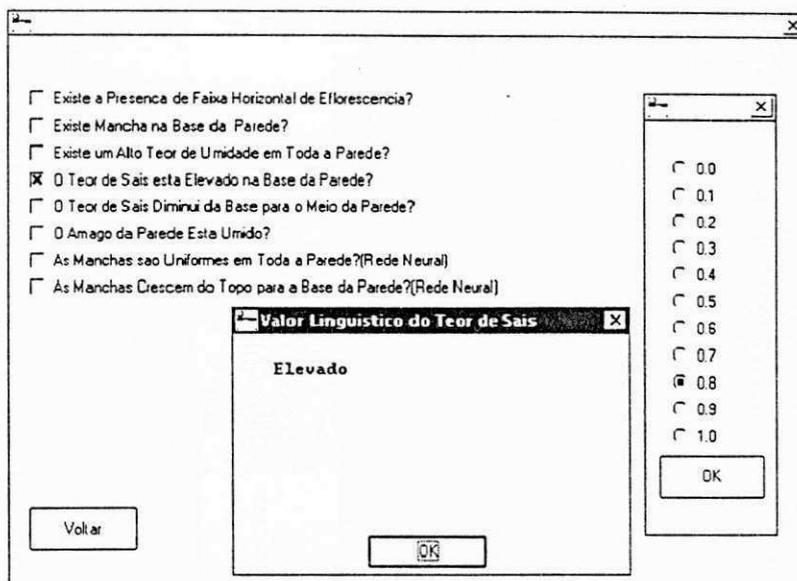


Fig. 5.20 - Tela do Resultado Fuzzy

No próximo capítulo serão apresentadas as conclusões desta dissertação de mestrado e sugestões para trabalhos a serem desenvolvidos.

## Capítulo 6

### CONCLUSÃO

A utilização dos sistemas híbridos neurosimbólicos que combinam conceitos de redes neurais, sistemas especialistas e lógica fuzzy constitui um grande avanço na área da Inteligência Artificial por integrar num só ambiente a força das redes neurais, o poder de explicação dos sistemas especialistas e a expressividade da lógica fuzzy. Esta nova abordagem híbrida é de qualidade superior, com utilização mais geral que as abordagens isoladas e favorece o desenvolvimento de sistemas inteligentes mais robustos e eficientes.

O SISNES, SISTEMA híbrido NEurosimbólico usando Subprocessamento, apresentado neste trabalho, é capaz de realizar o aprendizado através de exemplos e de regras simbólicas, permitindo a integração das abordagens simbolista e conexionista. O SISNES utiliza a lógica fuzzy no tratamento dos dados e na comunicação com o usuário.

Esta dissertação apresentou as classificações e arquiteturas dos sistemas híbridos neurosimbólicos e exemplos de sistemas existentes classificados conforme sua arquitetura e aplicação. Mostraram-se abordagens que tentam classificar os sistemas híbridos neurosimbólicos conforme alguns critérios e definições. De acordo com os critérios de Orsier[ORS 95A] classificamos o SISNES como uma abordagem híbrida com módulos separados, acoplamento estreito e subprocessamento. Nessa classificação a transferência de informação e controle ocorre de forma bidirecional, utiliza a memória como meio de

comunicação principal e arquivos em disco para o armazenamento dos dados de comunicação.

O SISNES possui explicação simbólica e conexionista. A explicação simbólica é baseada nas regras que foram utilizadas para determinar a solução final do sistema. A explicação conexionista é baseada na saída da rede neural. A explicação conexionista se responsabiliza por esclarecer o resultado da rede neural e foi implementada por uma falsa inversa da rede neural que foi utilizada. Através dos testes feitos com o SISNES a explicação conexionista se mostrou satisfatória.

Apresentaram-se resultados experimentais de uma implementação do SISNES na área de automação predial, mais especificamente no diagnóstico da umidade nas alvenarias de um edifício. Usou-se a lógica fuzzy na descrição do teor de sais na parede.

A inclusão da lógica fuzzy nos sistemas híbridos neurosimbólicos se mostrou eficiente no tratamento dos dados tornando a comunicação com o usuário mais clara. A lógica fuzzy permite a adaptação nas grandezas analisadas pelo SISNES para melhor entendimento do usuário. A declividade e o deslocamento das retas que representam as funções de pertinência podem ser modificados de acordo com a aplicação.

O SISNES é uma ferramenta que poderá ser facilmente adaptada para resolver outro problema que possuem características de subprocessamento. Quase nenhuma alteração será necessária na parte da integração entre os módulos simbolista e conexionista. As mudanças básicas necessárias no caso de utilizar o SISNES para outra aplicação são:

- Extração e elaboração das regras simbólicas relativas a aplicação. As regras deverão estar num arquivo.
- Criação e treinamento das redes neurais. As redes neurais deverão estar em um arquivo.
- Troca dos arquivos correspondentes às regras e as redes neurais do SISNES para os novos arquivos

Trabalhos que podem ser acrescentados a esta dissertação são:

- Fazer o produto final , cobrindo as etapas de planejamento e construção.

- Estudar outras situações do prédio em que é interessante o uso das redes neurais artificiais trabalhando em conjunto com regras simbolistas. O problema ideal seria aquele praticamente não resolvido por nenhuma das duas abordagens (simbolista e conexionista) isoladas e bem resolvido quando utilizada a abordagem híbrida neurosimbólica..
- Usar a rede neural de Hopfield para situações em que a otimização seja necessária. Por exemplo, para problemas do tipo caixeiro viajante.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [BEA 90] **BEALE R. & JACKSON T.** *Neural Computing: An Introduction*. Department of Computer Science, University of York. ed. Adam Hilger, Briston, Philadelphia and New York. 1990.
- [DON 94] **DONATO, E. T.** *Uso de Conhecimento Preliminar na Melhoria do Aprendizado em um Modelo Simbólico-Conexionista*. Dissertação de mestrado. Coordenação de Pós-Graduação em Informática - Universidade Federal da Paraíba, Campina Grande -PB.1994.
- [FAH 90] **FAHLMAN, S. & LEBIERE, C.** *The Cascade-Correlation Learning Architedure*. Carnegie Mellon University CMU-CS-90100.Feb.1990 Technical Report.  
(<ftp://reports.adm.cs.cmu.edu/usr0/anon/>)
- [FIS 89] **FISHER, D. H. & MCKUSICK, K. B.** *An Empirical Comparison of ID3 and Back-propagation*. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence. pp. 788-793. Detroit, 1989.
- [GAL 88] **GALLANT, S. I.** *Connectionist Expert Systems*. Communications of the ACM, vol. 31, pp. 152-169, 1988.
- [GAR 96] **GARRET JR, J. H. & SMITH, I. F. C.** *AI Applications in Structural/Construction Engineering*. IEEE Expert Intelligent Systems & Their Applications. vol. 11. Nº 3. pp. 20-52. 1996.
- [GIA 92] **GIACOMETTI, ARNAUD.** *Modelés Hybrides de l'Expertise*. Tese Doutorado - LIFIA/Telcom- Paris- France. 1992.  
(<http://lifia.imag.fr/RESEAUX/public.html>).
- [GOO 95] **GOOTNATILAKE S. & KHEBBAL S.** *Inteligent Hybrid Systems*. University College - London. London.1995.  
(<http://www.cs.ucl.ac.uk/staff/skhebbal/ihs/ihsbook.html>).
- [HAR 90] **HARMON P. & SAWYER B.** *Creating Expert Systems for Business and Industry*. ed. Jonh Willey & Sons. New York.1990.

- [HIL 94] **HILARIO M., PELLEGRINI C. & ALEXANDRE F.** *Modular Integration of Connectionist and Symbolic Processing in Knowledge-Based Systems*. International Symposium on Integrating Knowledge and Neural Heuristics. pp 123-132. Pensacola, Florida. 1994.  
(<http://www.loria.fr/exterieur/equipe/rfia/cortex/mix/mix.html>).
- [HIL 95] **HILARIO M., LALLEMENT Y. & ALEXANDRE F.** *Neurosymbolic Integration: Unifield versus Hybrid Approaches*. European Symposium on Artificial Neural Networks (ESBANN). Brussels. 1995.  
(<ftp://aramis.cs.ua.edu/pub/tech-reports/>)
- [HOP 82] **HOPFIELD J., J.** *Neural networks and physical systems with emergent collective computational abilities*. Proceedings of the National Academy of Sciences, 79:2554-2558.
- [HOP 84] **HOPFIELD J., J.** *Neurons with graded response have collective computational properties like those of two-state neurons*. Proceedings of the National Academy of Sciences, 81:3088-3092.
- [JUG 96] **JUGEND DAVID** *Automação Predial- Evolução e Comparação*. I Simpósio de Automática Aplicada. São Paulo - SP. 1996.
- [JUN 96] **JUNGMAN MILTON** *A Operação de Edifícios de Alta Tecnologia*. I Simpósio de Automática Aplicada. São Paulo - SP. 1996.
- [KAS 90] **KASABOV N. K.** *Hybrid Connectionist Rule-Based Systems*, Artificial Intelligence IV: Methodology, Systems, Applications, pp. 227-235, 1990.
- [KLE 88] **KLER, G. J. & FOLGER, T. A.** *Fuzzy Sets, Uncertainty and Information*, Prentice Hall, Englewood Cliffs, N. Jersey, 1988.
- [LAC 90] **LACERDA, A. M. C.** "Umidade Ascendente em Alvenarias Tradicionais". Dissertação de mestrado Faculdade de Arquitetura, UFBA, Salvador -Ba, 1990.
- [LOP 93] **LOPES, H. S.** *Arquitetura Híbrida Integrando Paradigmas Conexionistas e Simbolistas*. I SBAI - UNESP - Rio Claro-SP, 1993.

- [MOO 89] MOONEY, R., SHAVLIK, J., TOWELL, G. & GOVE, A. *An Experimental Comparison of Symbolic and Connectionist Learning Algorithms*. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence. pp. 775-780. Detroit, 1989.
- [ORS 94] ORSIER, B., RIALLE V. & GIACOMETTI A. *A Study of the Hybrid System SYNHESYS*. Workshop ECAI-94 "Combining Symbolic and Connectionist Processing." Amsterdam, 1994.  
(<http://cuiwww.unige.ch/AI-group/staff/orsier.html>)
- [ORS 95A] ORSIER, BRUNO. *Etude et Application de Systèmes Hybrides Neurosymboliques*. Tese Doutorado - LIFIA UJF - Paris - France. 1995.  
(<http://cuiwww.unige.ch/AI-group/staff/orsier.html>).
- [ORS 95B] ORSIER, BRUNO & LABBI ABDERRAHIM. *NESSY3L: NEuroSymbolic SYstem with 3 Levels*. IJCAI - Workshop on Connectionist-Symbolic Integration - Montreal - Canada. 1995.  
(<http://cuiwww.unige.ch/AI-group/staff/orsier.html>).
- [Oso 95] OSÓRIO, F. S. & AMY, B. *Um Sistema Híbrido Simboli-Conecionista com Aprendizado à Partir de Regras e de Exemplos*, PANEL'95 - XXI Conferência Latino Americana de Informática, vol. 2, pp. 509-520, Canela-RS, 1995.
- [RIC 94] RICH, E. & KNIGHT, K. *Inteligência Artificial* MAKRON Books do Brasil, São Paulo-SP, 1994.
- [ROS 57] ROSENBLATT, F. *"The Principles of Neurodynamics"*, Spartan 1962.
- [RUM 89] RUMELHART, D., HINTON, G. & WILLIAMS, R. *Parallel Distributed Processing*, Vol.1, MIT Press, Cambridge, 1989.
- [SCH 93] SCHIFMANN, W., JOOST, M. & WERNER, R. *Comparasion of Optimized Backpropagation Algorithm for Training Multilayer Perceptrons*. Proc. of the European Symposium on Artificial Neural Network, ESANN'93, Brussels, p.97-104.1993

- [SCH 95] SCHIFMANN, W.; JOOST, M. & WERNER, R. *Optimization of the Backpropagation Algorithm for Training Multilayer Perceptrons*. Tech. Report - University of Koblenz, Deutschland. Sept. 1995.  
(ftp://archive.cis.ohio-state.edu/pub/neuroprose/)
- [SHA 92] SHAVLIK, JUDE W. *A Framework for Combining Symbolic and Neural Learning*. Tech. Report.(1123). Dept. of Computer Science - University of Wisconsin - Madison. 1992.  
(ftp://ftp.cs.wise.edu/machine-learning/shavlik-group/)
- [SOU 94] SOUZA, A. H. C., ANDRADE, P. S. BITTENCOURT, C. T. *Um Sistema Especialista para Problemas de Umidade em Edificações*. XXVI Congresso Nacional de Informática e Telecomunicações, Salvador-Ba,1994.
- [SUN 93] SUN, RON. *An Efficient Feature-based Connectionist Inheritance Scheme*.IEEE Transactions on System, Man and Cybernetics. vol. 23. N° 2. pp. 512-522. 1993.  
(ftp://aramis.cs.ua.edu/pub/tech-reports/).
- [SUN 94] SUN, RON. *Robust Reasoning: Integrating Rule-Based and Similarity-Based Reasoning*.Tech. Report - Dept of Cumputer Science - University of Albama. 1994.  
(ftp://aramis.cs.ua.edu/pub/tech-reports/).
- [TOW 90] TOWWEL, GEOFFREV G. *Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks*. Eighth National Conference on Artificial Intelligence, pp.861-866. 1990.
- [TOW 91A] TOWWEL, GEOFFREV G. *Symbolic Knowledge and Neural Networks: Insertion, Refinament and Extraction*. PhD. Thesis - Univ of Wisconsin-Madison-USA.1991.  
(ftp://ftp.cs.wise.edu/machine-learning/shavlik-group/).

- [TOW 91B] TOWWEL, GEOFFREV G., CRAVEN, W. MARK & SHAVLIK, JUDE W. *Constructive Induction in Knowledge-Based Neural Networks*. Machine Learning: Proceedings of the Eighth International Workshop, California 1991.
- [TOW 94] TOWWEL, GEOFFREV G. & SHAVLIK, JUDE W. *Knowledge-Based Artificial Neural Networks*. Artificial Intelligence, vol. 70. 1994.  
(<ftp://ftp.cs.wise.edu/machine-learning/shavlik-group/>)
- [UTG 88] UTGOFF, P. E. *Perceptron Trees: A Case Study in Hybrid Concept Representations*. Proceedings of the Seventh National Conference on Artificial Intelligence. pp. 601-606. St. Paul, 1988.
- [ZAD 93] ZADEH, L. A. *Fuzzy Logic*. Paper 1.12 University of California, Berkeley, 1993.
- [ZIM 91] ZIMMERMANN, H. J. *Fuzzy Set Theory - and Its Applications*, 2nd ed., Kluwer Academic Publishers, 1991.