

XIX CONGRESSO DA S.B.C.
ENIA '99 – ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL

TUTORIAL

SISTEMAS

HÍBRIDOS

INTELIGENTES

Autores:

Prof. Dr. Fernando Santos Osório
Prof. Dra. Renata Vieira

UNISINOS – Universidade do Vale do Rio dos Sinos
Centro de Ciências Exatas e Tecnológicas – Centro 6/6
Curso de Informática
Web: <http://www.inf.unisinos.br>
E-Mail: osorio@exatas.unisinos.br
renata@exatas.unisinos.br

Rio de Janeiro, Julho 1999

SUMÁRIO:

0. Apresentação
1. Inteligência Artificial: Métodos Simbólicos (R. Vieira)
2. Inteligência Artificial: Métodos Conexionistas – Rede Neurais (F. Osório)
3. Sistemas Híbridos Inteligentes (F. Osório)

APRESENTAÇÃO

Este tutorial tem por objetivo apresentar os conceitos e as novas tendências da área de *Sistemas Híbridos Inteligentes*. Serão abordados os métodos tradicionais simbólicos e conexionistas, apontando para os principais problemas e limitações de cada tipo de técnica em aplicações do mundo real. A necessidade do tratamento de grande quantidade de informações e as incertezas inerentes aos sistemas complexos nos leva a fusão destas metodologias, representada pelos Sistemas Híbridos.

Inicialmente nós apresentaremos as características dos diferentes métodos de representação do conhecimento e suas particularidades. Discutiremos a utilização destas técnicas no desenvolvimento de sistemas inteligentes (sistemas baseados em conhecimento: lógica, regras, casos, frames, redes semânticas) e suas limitações na aquisição do conhecimento e na representação de conhecimento incerto e incompleto.

Nesta seção vamos analisar os métodos conexionistas (redes neurais). Inicialmente será apresentada uma visão geral sobre as redes neurais, os diferentes tipos de redes e algoritmos de aprendizado existentes. Serão discutidas as características e limitações de uso dos diferentes modelos conexionistas, relacionando-as com as características dos sistemas de inteligência artificial simbólica vistos na seção anterior.

O problema da aquisição e na representação do conhecimento incompleto e incerto impulsiona a fusão de técnicas e metodologias, originando os sistemas híbridos. Serão apresentados os conceitos básicos relativos aos sistemas híbridos inteligentes. Vamos apresentar uma visão global das possíveis combinações entre sistemas de Inteligência Artificial: Redes Neurais, Algoritmos Genéticos, Lógica Nebulosa, Sistemas baseados em Conhecimento, Sistemas Baseados em Casos, etc. Discutiremos sobre os modos de integração destes sistemas: processamento em série, sub-processamento, meta-processamento e co-processamento. A partir de uma análise das vantagens da integração destes diferentes métodos, vamos direcionar nossa apresentação ao que se refere aos sistemas híbridos neuro-simbólicos. Em seguida, mostraremos exemplos de sistemas híbridos neuro-simbólicos, tais como: KBANN, SYNHESYS e INSS (ligado ao projeto Europeu Esprit/BRA MIX). Concluiremos esta seção com uma análise sobre os sistemas inteligentes e o uso de ferramentas híbridas.

1.INTELIGÊNCIA ARTIFICIAL:

Métodos Simbólicos

1.1 Introdução

Os métodos simbólicos em Inteligência Artificial são baseados na idéia de que a inteligência (ou parte dela) constitui-se de operações sobre um conjunto de símbolos. A linguagem, exemplo de sistema simbólico, é uma faculdade humana associada à inteligência. As palavras representam objetos, relações e eventos no mundo (são símbolos). Quando combinadas umas com as outras podem descrever, explicar (ou influenciar) diversas situações. Podemos entender uma situação através de sua representação simbólica, sem necessariamente vivenciá-la e utilizar este conhecimento posteriormente em situações reais.

Os primeiros sistemas de inteligência artificial foram os sistemas resolvedores de problemas. A programação deste tipo de sistemas requer uma especificação precisa do problema em termos de estados e operações sobre estes estados. A solução de um problema é dada ao atingir-se o estado final especificado a partir da aplicação de um conjunto de operações sobre o estado inicial. O conjunto de passos a serem realizados do estado inicial ao final constitui a solução do problema. A especificação do problema é uma representação simbólica, fornecida de tal maneira que um algoritmo seja capaz de operar sobre esta representação e reproduzir uma representação simbólica da solução.

Os métodos ditos simbólicos em Inteligência Artificial seguem este mesmo princípio, são baseados em técnicas de representação simbólica de conhecimento associadas a mecanismos com capacidade de inferir soluções ou novos conhecimentos a partir do conhecimento representado.

Neste capítulo iremos apresentar idéias básicas sobre representação do conhecimento, a representação simbólica e sistemas que manipulam representações simbólicas, no desempenho de funções normalmente associadas à inteligência humana.

1.2 Representação do conhecimento

Uma representação é alguma coisa que está no lugar de outra. É um modelo da coisa representada, de maneira que olhando-se apenas para o modelo, conclusões podem ser tiradas sobre as coisas representadas. Para o mesmo objeto sendo representado podem corresponder vários modelos representantes. Veja a Tabela 1 [Rumelhart 84], tem-se um mundo representado (indivíduos, propriedades e relações) e três modelos representantes.

Mundo representado	Modelos representantes		
	Modelo 1	Modelo 2	Modelo 3
Objetos			
Indivíduo 1	A		1,80
Indivíduo 2	B		1,50
Propriedades			
Altura		comprimento de linha	valor numérico
Relações			
Maior que	mais_alto_que(A,B)	mais comprida	>

Tabela 1. Diferentes modelos de representação.

No primeiro os indivíduos são representados pelos símbolos A e B, a relação de tamanho entre os dois indivíduos é representada pela fórmula *mais_alto_que(A,B)*. Não há representação direta de altura neste sistema. No segundo modelo, os indivíduos são representados por linhas verticais, a relação de tamanho é representada pela relação física entre as duas linhas. No terceiro modelo, os indivíduos são representados por números e a magnitude

dos números representam suas respectivas alturas. A relação mais alto que neste caso é representada pela relação aritmética $<$ (maior que). Estas diferentes representações estão associadas a diferentes processos de interpretação e uso: se a altura é representada pelo comprimento da linha, existe um processo para comparação de comprimentos de linha, se a altura é representada por números, deverá corresponder um processo capaz de operar sobre estes números.

Existem diferentes sistemas de representação. Existem também sistemas híbridos, que combinam dois ou mais diferentes sistemas. Alguns sistemas típicos são descritos brevemente a seguir.

Sistema proposicional - em um sistema proposicional os conceitos são representados por afirmações. É geralmente utilizado para representar conhecimento declarativo. Neste tipo de representação o conhecimento pode ser examinado e combinado, produzindo inferências.

Sistema analógico - a representação é feita por correspondência direta, um mapa é um exemplo de sistema analógico de representação de características geográficas.

Sistema procedural - este tipo de representação não é acessível, por isto é difícil de ser examinado. Um sistema procedural de representação é um *know-how*, utilizados por exemplo na execução de determinadas ações (como chutar uma bola ou andar de bicicleta).

Em Inteligência Artificial Simbólica, a representação do conhecimento é proposicional, é feita principalmente através de uma coleção estruturada de símbolos e com procedimentos de interpretação.

1.3 Representação simbólica do conhecimento

Diferentes técnicas de representação simbólica do conhecimento foram propostas, atendendo a necessidades e aplicações específicas. Estas técnicas são usadas para expressar o conhecimento de forma que possa ser tratado computacionalmente.

1.3.1 Lógica

O desenvolvimento da lógica tem origem na Grécia antiga. Buscava-se uma coerência entre a estrutura de sentenças ou conjuntos de sentenças e a verdade. No final do século XVII Leibniz propõe a criação de uma linguagem lógica formal procurando reduzir a inferência lógica a um processo mecânico. Neste processo de formalização da lógica participaram nomes conhecidos como: Boole, Frege e Gödel. O desenvolvimento da lógica foi fundamental para o desenvolvimento da Inteligência Artificial Simbólica. Uma linguagem precisa com um processo de inferência bem definido pode ser utilizada para representar conhecimento sobre fatos e derivar mecanicamente suas conseqüências. Por outro lado, ao fazer uso da lógica para expressar conhecimento e simular o raciocínio humano, a Inteligência Artificial lançou novos desafios e estimulou o desenvolvimento de lógicas que pudessem tratar exceções, inconsistências, conhecimento incerto e incompleto, as lógicas não-clássicas. As lógicas para a representação do conhecimento incluem: lógica modal, lógicas não monotônica (lógica default, autoepistêmica e circunscrição), lógica difusa, raciocínio sobre ação e mudança (cálculo de situação, lógica temporal), raciocínio sobre estados mentais (crenças, obrigações, objetivos).

Uma linguagem lógica é definida por sua sintaxe e semântica. A sintaxe descreve as configurações possíveis das sentenças da linguagem e a semântica descreve a relação das sentenças com fatos no mundo. Além da linguagem é necessário definir um conjunto de regras usadas para derivar novas sentenças a partir de sentenças dadas, estas regras definem o mecanismo de inferência.

Através de uma linguagem lógica clássica de primeira ordem pode-se representar um domínio constituído de objetos ou entidades, expressar propriedades destes objetos (específicos = constantes da linguagem; genéricos = variáveis da lógica) e relações entre eles. Estas propriedades e relações podem ser quantificadas (valer para todos \forall ou para alguns \exists). As relações entre diferentes fatos ou proposições são expressas por conectivos lógicos (conjunção \wedge , disjunção \vee , consequência \rightarrow e negação \neg). A seguinte sentença lógica pode ser usada para expressar o fato de que pássaros são animais:

$$\forall x \text{ passaro}(x) \rightarrow \text{animal}(x)$$

Geralmente a representação de um domínio um pouco mais complexo em lógica de primeira ordem não pode ser lido e interpretado de uma maneira natural e intuitiva. Isto motivou o desenvolvimento de outros métodos de representação de conhecimento, que serão descritos a seguir. Estes outros métodos oferecem um formato que é mais claro ao leitor humano e facilitam o processo de traduzir o conhecimento, usualmente expresso em linguagem natural, para uma linguagem mais formal e estruturada. Estes métodos, porém, podem ser traduzidos em uma notação lógica. Basicamente eles apresentam uma sintaxe diferente, preservando semântica e teoria da prova.

1.3.2 Redes semânticas

Em 1968, Quillian introduz as redes semânticas, uma estrutura de representação de conhecimento na forma de grafo estruturado formado por um conjunto de nós interligados por relações (Figura 1).

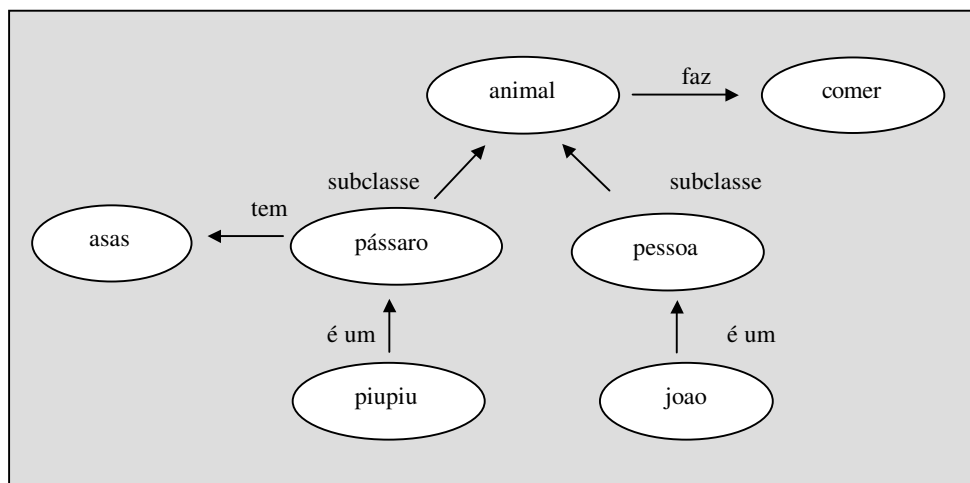


Figura 1. Redes semânticas

As redes semânticas procuram representar os significados das palavras, tidos como conceitos inter-relacionados na memória. Os nós da rede representam os conceitos, a relação é uma associação entre estes. A relação *é um* entre dois nós indica pertinência de uma instância a uma classe, a relação *subclasse* representa relação de pertinência entre classes.

As redes semânticas introduziram a idéia de herança de atributos, cada instância ou subclasse herda os atributos válidos para a classe. Outras relações podem ser representadas, tais como, *tem*, *faz*, *parte*, etc.

A ativação de uma rede é uma busca por uma relação entre nós, através das relações. Esta busca pode ser utilizada para extrair informações. Por exemplo, a proposição *piupiu tem asas* pode ser encontrada na rede, através do caminho: *piupiu é um pássaro, pássaros têm asas*. Todas as informações sobre *piupiu* podem ser fornecidas a partir de uma busca exaustiva na rede.

A notação apresenta uma representação visual intuitiva, e proporciona flexibilidade de manutenção (modificação). Apesar de ter sido muito influente no desenvolvimento da representação do conhecimento e largamente usada para especificar conhecimento de domínios mais restrito, torna-se confusa para redes mais abrangentes e complexas.

Enquanto as redes semânticas relacionam palavras e conceitos, outras técnicas surgiram para tratar unidades maiores do conhecimento, exemplos são os *Frames* (ou Quadros) [Minsky, 1975]; *Scripts* [Schank, 1977].

1.3.3 Frames

Frames (ou quadros) são um método de representação no qual uma classe é definida por um conjunto de atributos (*slots*) com determinados valores. Os valores dos atributos podem ser eles mesmos outros *frames*. São também referenciados como estruturas de escaninho e preenchimento (*slot-and-filler*). São equivalentes a redes semânticas, porém são mais estruturados. Assim como as redes, *frames* são organizados em uma hierarquia de especialização que fornece uma estrutura natural para herança de atributos. Relações do tipo *é um* e *subclasse* ligam objetos e classes a outras classes e propriedades dos objetos ou subclasses podem ser determinadas no nível do objeto, da subclasse ou da superclasse. Esta estrutura é adequada para representar a idéia de valores *default*. Os *frames* podem indicar valores específicos para os atributos, ou então indicar valores *default*. Isto é útil para representar valores válidos na maioria dos casos em uma classe, mas que podem assumir um valor diferente em uma especialização da classe. Por exemplo a classe pássaro pode ter com valor *default* a habilidade de voar, uma subclasse específica (avestruz) podem ter esta habilidade cancelada.

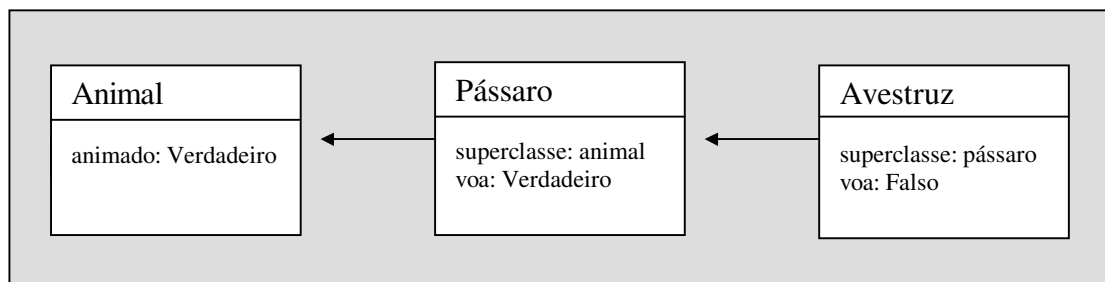


Figura 2. *Frames*

Frames são eficientes em domínios que apresentam uma taxonomia natural de conceitos, como por exemplo, a biologia. Os atributos principais de um frame são nome e a especificação de classe (relação *é um* ou *subclasse*). Os atributos podem ser procedimentos a serem executados para a obtenção de algum valor, quando necessário.

1.3.4 Scripts

Um *script* é um esquema para uma seqüência de eventos que compõem uma situação prototípica como ir ao restaurante, ir ao médico, viajar de trem. O *script* representa o conhecimento sobre determinadas situações usuais, uma vez representado, um *script* pode ser utilizado para auxiliar no entendimento da ordem de eventos. Em um segmento de discurso do tipo:

Maria foi ao restaurante. Pediu uma torta, pagou e saiu.

Podemos inferir através do conhecimento tipo *script* que ela estava com fome, tinha algum dinheiro, comeu a torta antes de pagar e que um garçom lá estava e a atendeu. O *script Restaurante* é um exemplo clássico da literatura de Inteligência Artificial, e está exemplificado na Figura 3. *Scripts* são compostos de objetos, agentes, condições de entrada, resultados e uma seqüência de cenas.

<p>Nome: Restaurante Objetos: mesas, menu, comida, conta, dinheiro Agentes: cliente, garçom, cozinheiro, caixa, dono Condições de entrada: o cliente tem fome, o cliente tem dinheiro Resultados: o cliente tem menos dinheiro, o dono tem mais dinheiro, cliente não tem fome</p>
<p>Cena 1: Entrada O cliente entra no restaurante O cliente procura por uma mesa O cliente decide onde sentar O cliente vai para a mesa O cliente senta</p> <p>Cena 2: Pedido O cliente pega o menu O cliente olha o menu O cliente decide o que vai comer O cliente chama o garçom O garçom vem até a mesa O cliente pede a comida O garçom vai até a cozinheiro O garçom entrega o pedido para o cozinheiro O cozinheiro prepara a comida</p> <p>Cena 3: Refeição O cozinheiro dá a comida ao garçom</p> <p>Cena 4: Saída</p>

Figura 3. Script Restaurante

1.3.5 Sistemas simbólicos híbridos

As vezes, diferentes métodos de representação são mais ou menos adequados dependendo da aplicação. Regras e *frames*, por exemplo, são dois esquemas bem distintos. Regras são apropriadas para a representação de implicação lógica e associação de ações com condições sobre as quais ações deveriam ser aplicadas. *Frames* e redes semânticas são apropriadas para definir termos e descrever objetos e relações taxonômicas entre eles (membro/classe). Por serem muitas vezes complementares, alguns sistemas empregam uma combinação de métodos de representação.

1.4 Sistemas simbólicos inteligentes

A evolução da Inteligência Artificial simbólica passou por propostas de diferentes sistemas. Uma breve apresentação desta evolução é dada nesta seção. Os problemas e as técnicas desenvolvidas para tratá-los crescem em complexidade, mas todas adotam a perspectiva de representação simbólica do conhecimento: operação sobre símbolos e produção de um resultado simbólico significativo e interpretável pelos usuários destes sistemas.

Alguns destes sistemas foram comercialmente viáveis e possibilitaram uma popularização da área de Inteligência Artificial. O desenvolvimento destes sistemas é paralelo ao desenvolvimento das técnicas de representação de conhecimento apresentadas na seção anterior.

1.4.1 Resolução de problemas

A construção de sistemas de resolução de problemas baseia-se na descrição formal e manipulável do problema, onde estados possíveis e operações de mudanças de estados são especificados. Apresentaremos como exemplo o problema das jarras de água. O enunciado do problema é seguinte: Você tem 2 jarras, uma de 4 litros, outra de 3 litros. Você não pode ver o conteúdo das jarras e tem que encontrar uma maneira de colocar 2 litros na jarra de 4 litros. A Figura 4 apresenta um exemplo de especificação para este problema.

Estados:	{ (0,0) , (0,...) , (0,3) , ... (4,3) }	
Estado inicial:	(0,0)	
Objetivo:	(2,n)	
Operações: (ações):	pré-condição	efeito
1. Encher J4 (x,y)	(x<4)	(4,y)
2. Encher J3 (x,y)	(y<3)	(x,3)
3. Esvaziar J4 no chão (x,y)	(x>0)	(0,y)
4. Esvaziar J3 no chão (x,y)	(y>0)	(x,0)
5. Despejar J3 em J4 e encher J4 (x,y)	(x+y>4, y>0)	(4,y-(4-x))
6. Despejar J4 em J3 e encher J3 (x,y)	(x+y>3, x>0)	(x-(3-y),3)
7. Despejar toda J3 em J4 (x,y)	(x+y<3, x>0)	(0,x+y)

Figura 4. Especificação do problema das jarras de água

A partir desta especificação, encontrar a solução de um problema é buscar uma seqüência finita de operações que levem ao estado final. Para a busca de soluções em espaços de estados foram desenvolvidas estratégias de navegação em estruturas do tipo árvore, onde os estados são representados por nós, a raiz é o estado inicial e o desdobramento da árvore representa o caminho já trilhado na busca da solução.

1.4.2 Planejamento

Algoritmos de planejamento são descritos por formalismos lógicos. Estados e objetivos são representados por um conjunto de sentenças e ações são representadas por uma descrição lógica de suas pré-condições e efeitos. Uma solução é um plano de ações que garantem a realização de um objetivo.

Um exemplo clássico de planejamento é dado através do mundo dos blocos. O domínio do problema é um conjunto de blocos sobre uma mesa. Os blocos são representados por constantes e a posição relativa dos blocos é dado pela relação *Sobre(x,y)*. Os blocos podem ser empilhados, e um braço de robô pode mover estes blocos sobre a mesa distribuindo-os em pilhas. Um objetivo é uma disposição de blocos sobre a mesa e uns sobre outros, por exemplo, formar duas pilhas, uma com o bloco A sobre o B e outra com bloco C sobre D. Um plano é uma seqüência de ações a serem realizadas para atingir o objetivo.

A realização de ações deve respeitar pré-condições e produzir efeitos. Por exemplo, a ação mover um bloco só pode ocorrer quando não há nenhum bloco sobre o bloco a ser movimentado e o efeito é uma nova relação *Sobre(x,z)*.

1.4.3 Jogos

Jogos de tabuleiro são constituídos por um conjunto específico de ações, relativamente fáceis de serem representadas, e são, portanto, uma aplicação natural para as técnicas desenvolvidas pela Inteligência Artificial. Podem ser representados como um problema com estado inicial, um conjunto de operadores e estado final. Ações do adversário devem ser consideradas e o melhor movimento deve ser calculado a fim de garantir a vitória.

1.4.4 Sistemas de produção

Sistemas de produção caracterizam uma arquitetura genérica que pode ser adotada por diferentes sistemas (jogador de xadrez, solucionador do problema das jarras de água, etc.). Esta arquitetura é formada por

- um conjunto de regras (se LE então LR) com lado esquerdo (LE) e direito (LR), indicando respectivamente aplicabilidade e operacionalidade da regra;
- uma base de conhecimento (conhecimentos permanentes ou temporários)
- uma estratégia de controle que especifica a ordem de aplicação das regras (ou motor de inferência)

1.4.5 Sistemas especialistas

Os sistemas especialistas são responsáveis pela re-descoberta e popularização da Inteligência Artificial, obtiveram grande êxito em domínios específicos onde o conhecimento pudesse ser traduzido em regras do tipo causa e consequência, ou sintomas e diagnóstico.

Uma vez estruturado e formalizado, o conhecimento de uma área específica pode ser manipulado e reutilizado. Uma determinada situação observada pelo usuário e informada ao sistema, pode ser indicadora de uma nova situação e informada ao usuário pelo sistema. Por exemplo, uma determinada doença pode ser diagnosticada a partir da informação de certos sintomas observados. As razões para a conclusão apresentada podem ser explicadas ao usuário.

Um sistema especialista possui a mesma arquitetura básica de um sistema de produção. A construção da base de conhecimento é bastante elaborada e passa pelas fases de aquisição, representação, formalização, codificação, refinamento e validação. A base de conhecimento possui conhecimento ontológico (tipo de conhecimento descrito pelas redes semânticas), regras e fatos. A memória temporária registra informações a respeito de um caso particular sendo operado pelo sistema. O motor de inferência é responsável pela geração das conclusões.

Os sistemas especialistas são geralmente aplicados em domínios sem qualquer teoria geral, suscetíveis a descrições incompletas ou incertas. Alguns métodos para tratamento da incerteza foram propostos, tais como: método dos fatores de certeza, teoria dos conjuntos nebulosos, teoria de probabilidades e de possibilidades. Geralmente estes métodos atribuem uma medida numérica que representa a confiança do especialista nos fatos.

No método de fatores de certeza, por exemplo, cada regra e cada elemento da memória de trabalho são associados a um coeficiente de certeza, este que pode variar entre -1 (totalmente falso) e 1 (totalmente verdadeiro). Estes coeficientes são utilizados para propagar a incerteza inicial de uma informação através da cadeia de inferências

Existem sistemas que deixam o usuário optar pelo método que deseja ser aplicado, pois cada método tem uma aplicabilidade melhor ou pior de acordo com os problemas ou domínios. Uma característica comum a todos os métodos é a existência de um limite mínimo para a incerteza, abaixo deste a regra é desconsiderada.

Existem sistemas especialistas que, no caso de atingir um estágio de raciocínio incerto, interagem com o usuário solicitando informações para continuar o raciocínio. Além de se utilizarem das informações fornecidas pelo usuário para elaborar uma resposta ao problema, os sistemas especialistas devem ser capazes de adquirir novos conhecimentos ou modificar os antigos, podendo utilizar a mesma interface com o usuário.

Apesar de aplicarem bons métodos de tratamento de incerteza, nem sempre os sistemas especialistas conseguem encontrar as mesmas soluções que poderiam ser encontradas pelos especialistas humanos, pois muitas vezes as soluções humanas têm base intuitivas. Dificilmente os motivos da intuição dos humanos são colocados em regras na base de conhecimento.

1.4.6 Raciocínio baseado em casos

O método de raciocínio baseado em casos é baseado na idéia de que humanos usam o raciocínio analógico, baseado em experiências para aprender e resolver problemas. Este método é particularmente útil quando há informação incompleta. Casos consistem de

- a) informação sobre situações,
- b) as soluções adotadas para cada uma das situações,
- c) os resultados da utilização de determinada solução, e
- d) atributos utilizados para auxiliar na busca por casos semelhantes.

Os elementos de um sistema baseado em raciocínio são:

- a base de casos
- índice (usado para otimizar a procura e recuperação de casos apropriados ao problema sendo tratado)
- métricas de similaridade - utilizadas para medir a semelhança entre casos
- módulo de adaptação - responsável pela criação da solução do problema em questão através da modificação de solução adotada em problema semelhante (adaptação estrutural) ou criando uma solução nova utilizando o mesmo processo utilizado em casos anteriores (adaptação derivada).

1.4.7 Aprendizado

Existem diversas técnicas de aprendizagem simbólica. Alguns tipos serão citados, e o aprendizado indutivo, mais conhecido, será brevemente apresentado. A idéia geral em aprendizagem simbólica é de gerar um conjunto de regras ou um procedimento classificatório a partir de uma amostra de casos. Tipos de aprendizado são: a) aprendizado por analogia em Sistemas CBR - *Case Based Reasoning*; b) aprendizado por indução em ILP - *Inductive Logic Programming*; c) aprendizado baseado em explicação EBL - *Explanation-Based Learning*; d) aprendizado por evolução/seleção em GA e GP - *Genetic Algorithms / Genetic Programming*; e) o mais difundido, aprendizado indutivo a partir de exemplos ou Árvores de Decisão IDT - *Induction of Decision Tree*.

As árvores de decisão (*decision trees*) propostas por Quinlan [Quinlan 92] são um exemplo clássico de aprendizado indutivo. Neste método, um conjunto de casos, combinando uma série de características com determinada classificação é fornecido a um algoritmo que produz uma árvore de decisão a ser utilizada em novos casos. A árvore de decisão resultante dispõe as características observadas nos exemplos de maneira a otimizar as classificações futuras. O nodo inicial da árvore representa a primeira característica a ser analisada. A presença ou ausência da característica determina o nodo seguinte (esquerdo ou direito de uma árvore) que pode representar uma nova característica a ser analisada ou uma classificação resultante.

Uma outra área em Ciência da Computação relacionada ao aprendizado, que está em expansão, é a pesquisa em extração de informação. Esta tem por objetivo o desenvolvimento de métodos e sistemas para a aquisição automática de bases de conhecimento. A partir da leitura de textos em linguagem natural, deve ser produzida uma representação mais formal do conhecimento de um domínio específico.

1.4.8 Recentes desenvolvimentos: sistemas multi-agentes

Uma área mais recente da Inteligência Artificial Simbólica é a de especificação formal e projeto de agentes autônomos. Programas ou sistemas são entendidos como agentes que raciocinam logicamente, constroem e executam planos, são capazes de aprender e podem agir com autonomia (sem interferência ou solicitação direta do usuário).

Além da modelagem de um único agente, expande-se a pesquisa sobre times de agentes (dois ou mais cooperando em uma tarefa complexa). Para tarefas como esta é preciso não só lidar com a representação, mas com o reconhecimento e antecipação de estados mentais. Estes sistemas integram representação de crenças, planos, objetivos, desejos e intenções. A representação da emoção também tem sido considerada por pesquisadores da área.

1.5 Problemas da Inteligência Artificial Simbólica

Programar um conjunto de critérios para abranger os membros de uma categoria é uma tarefa extremamente complexa. A definição de solteiro, por exemplo, pode ser dada por: *um homem de idade adulta que não é casado*. No entanto, se um amigo pede para você convidar alguns solteiros para ir a sua festa, a definição acima provavelmente não seria adequada o suficiente para saber quem convidar das pessoas abaixo:

- um amigo não casado oficialmente mas que tem uma companheira e filhos,
- um amigo que casou para não ser convocado pelo exército, mas que nunca viveu com a esposa,
- um amigo homossexual.

Um sistema inteligente é alimentado com representações de fatos e regras, mas as regras do senso comum são muito difíceis de estabelecer. Sabe-se que quando o cachorro vai para o carro, ele não está mais no quintal; se alguém vai ao supermercado, sua cabeça vai junto; se alguém está dentro de casa deve ter entrado por alguma porta; abrir um pote de manteiga não encherá a casa de vapor; se há uma sacola em seu carro e um litro de leite na sacola então há um litro de leite em seu carro. Note porém que de saber que há uma pessoa em seu carro e saber que há um litro de sangue em uma pessoa seria estranho concluir que há um litro de sangue em seu carro.

O problema de saber distinguir as implicações relevantes entre um conjunto potencialmente infinito de implicações é denominado *frame problem* (problema de enquadramento), e dificulta principalmente a formalização de raciocínios sobre ações.

1.6 Conclusão

A Inteligência Artificial Simbólica sempre esteve intimamente ligada aos estudos da mente humana. Seus avanços foram sucintamente esboçados neste capítulo. Esbarrou em problemas relacionados ao conhecimento intuitivo, incerto e incompleto. Encontrou maiores dificuldades em representar o senso comum. Está longe a possibilidade de se apresentar um sistema para competir com a habilidade humana de lidar com a ambigüidade e fazer dela fonte de graça, ironia ou poesia.

Inicialmente a IA Simbólica mostrou relações mais fortes com a psicologia. Ultimamente, a área tem avançado suas relações com outras áreas ligadas à compreensão do comportamento humano e seu contexto (tais como a sociologia e a antropologia). Recentes avanços trazidos pelo paradigma multi-agente têm mostrado grande potencial para auxiliar nos estudos dos processos sócio-cognitivos. Ao mesmo tempo esta inter-relação com as áreas humanas inspira o trabalho da Inteligência Artificial Simbólica rumo a um novo patamar de competência dos sistemas computacionais.

Bibliografia

[Bittencourt 98] BITTENCOURT, Guilherme. Inteligência Artificial, Ferramentas e Teorias. Editora da UFSC, 1998.

[Bower 79] BOWER, G. H., BLACK, J. B. and TURNER, T. J. Scripts in memory for text. *Cognitive Psychology*, 11, 177-220. 1979.

[Davis 93] DAVIS, R., SHROBE, H. and SZOLOVITS P. What is a Knowledge Representation? AI Magazine, 14(1):17-33, 1993. <http://medg.lcs.mit.edu/ftp/psz/k-rep.html>

[Green 96] GREEN, David et al. *Cognitive Science: an introduction*. Cambridge, Blackwell Publishers, 1996.

[Minsky 75] MINSKY, Marvin. A framework for representing knowledge. In P. Winston (Ed.) *The psychology of computer vision*. New York: McGraw Hill.

[Minsky 91] MINSKY, Marvin. Logical vs. Analogical or Symbolic vs Connectionis or Neat vs. Scruffy, in *Artificial Intelligence Magazine*, 1991.

[Pinker 97] PINKER, Steven. *How the mind works*. W.W. Norton & Company. 1997.

[Quinlan 92] QUINLAN, J. R. *C 4.5 : Programs for Machine Learning* (Morgan Kaufmann Series in Machine Learning). Academic Press/Morgan Kaufmann,1992.

[Quillian 68] QUILLIAN, M. R. Semantic memory. In Minsky, M. L. Editor, *Semantic Information Processing*, pages 216-270. MIT Press, Cambridge, Massachusetts.

[Rich & Knight] RICH, Elaine; KNIGHT, Kevin. *Inteligência Artificial*. Segunda Edição. Editora McGraw-Hill Ltda., 1993.

[Russel 95] RUSSEL, Stuart; NORVIG, Peter. *Artificial Intelligence. A Modern Approach*. Prentice-Hall Inc, 1995.

[Rumelhart 84] RUMELHART, David & NORMAN, Donald. Representation of knowledge. In Aitkenhead M., Editor, *Issues in Cognitive Modelling*, 1984.

[Schank, 1977] SHANK R. C. & ABELSON R. P. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates, Potomac, Maryland. 1977.

[Wermter 96] WERMTER, Stefan; RILOFF, Ellen and SCHELER, Gabriele. *Connectionist, statistical and symbolic approaches to natural language processing*. Springer, Heidelberg, 1996.

2. INTELIGÊNCIA ARTIFICIAL : **Métodos Conexionistas - Redes Neurais Artificiais**

2.1. Introdução

As Redes Neurais Artificiais (RNA), também conhecidas como métodos conexionistas, são inspiradas nos estudos da maneira como se organiza e como funciona o cérebro humano. Este tipo de método possui características peculiares de representação e de aquisição de conhecimentos, sendo considerado um método de nível sub-simbólico (em oposição aos métodos simbólicos, descritos no capítulo anterior: árvores de decisão, CBR, KBS, etc).

Inicialmente vamos discutir sobre a representação de conhecimentos utilizada pelas Redes Neurais, para depois analisarmos a parte referente ao aprendizado destas. É importante salientar que existem diferentes tipos de RNAs e que cada um destes modelos diferentes possui suas características próprias quanto a forma de representar e de adquirir conhecimentos. Em função disto vamos primeiramente apresentar uma visão geral, classificando os diferentes modelos de RNAs para em seguida nos concentrarmos em um modelo mais específico: as redes neurais artificiais do tipo multi-nível baseada em Perceptrons (conhecidas como Multi-Layer Perceptron Nets ou Back-Propagation Nets). A figura 2.1 apresenta um exemplo deste modelo de rede conexionista.

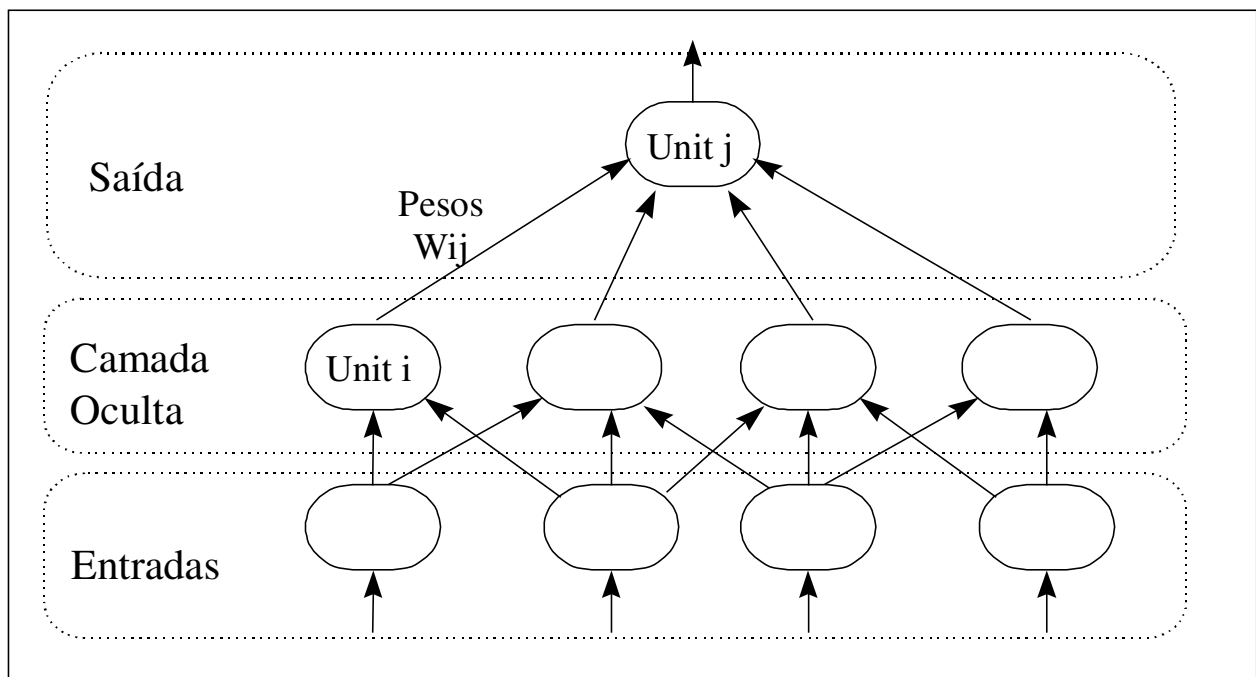


Figura 2.1. Exemplo de Rede Neural Artificial do tipo Multi-Nível

2.2. Representação de Conhecimento

A representação de conhecimentos nas redes conexionistas, como diz o próprio nome, é fortemente ligada a noção de conexão entre neurônios (elementos processadores de informação) que interagem uns com os outros através destas ligações. O modelo conexionista possui sua origem nos estudos feitos sobre as estruturas de nosso cérebro – *sofrendo uma grande simplificação do modelo original* – onde encontramos no modelo artificial, que é simulado, elementos como os neurônios e as suas conexões, chamadas de sinapses. A proximidade e fidelidade dos modelos artificiais em relação ao modelo real é um *tema polêmico* que não vamos nos aventurar a discutir aqui, deixando apenas o registro da origem dos conceitos básicos que norteiam este campo de pesquisas.

O conhecimento de uma RNA está codificado na estrutura da rede, onde se destacam as conexões (sinapses) entre as unidades (neurônios) que a compõe. Nestas redes artificiais obtidas por simulação em computadores, associamos a cada conexão um peso sináptico (valor numérico) que caracteriza a força da conexão entre dois neurônios. O aprendizado em uma RNA é realizado por um processo de adaptação dos seus pesos sinápticos. As figuras 2.2 e 2.3 mostram a relação entre os neurônios naturais e o modelo de neurônio artificial.

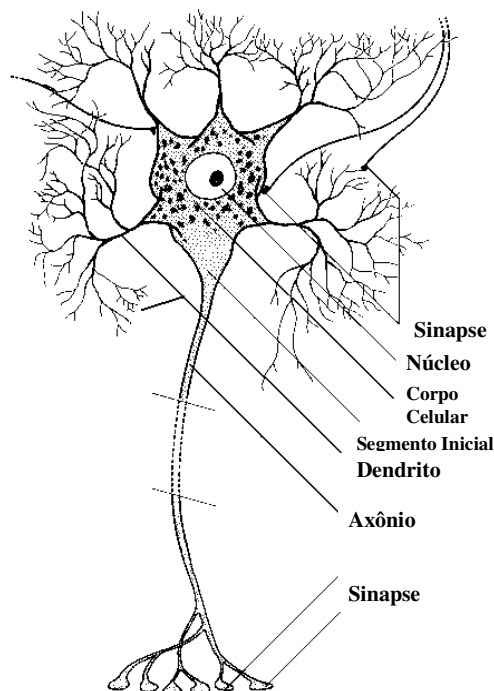


Figura 2.2. Exemplo de Neurônio Natural

Uma vez que os conhecimentos da rede estão codificados na estrutura de interconexões entre os neurônios e nos pesos associados a estas conexões, fica muito difícil para um ser humano realizar uma análise e interpretação dos conhecimentos adquiridos por uma RNA. Os conhecimentos das redes se resumem a um conjunto de valores numéricos descrevendo as conexões, e por consequência, estes valores descrevem também o comportamento da rede. Entretanto, para um ser humano estes dados não fazem muito sentido. Na seção 2.3 vamos apresentar mais em detalhes os diferentes tipos de redes, onde a escolha do tipo de neurônio artificial utilizado é fundamental para se definir como será a representação interna dos conhecimentos da rede.

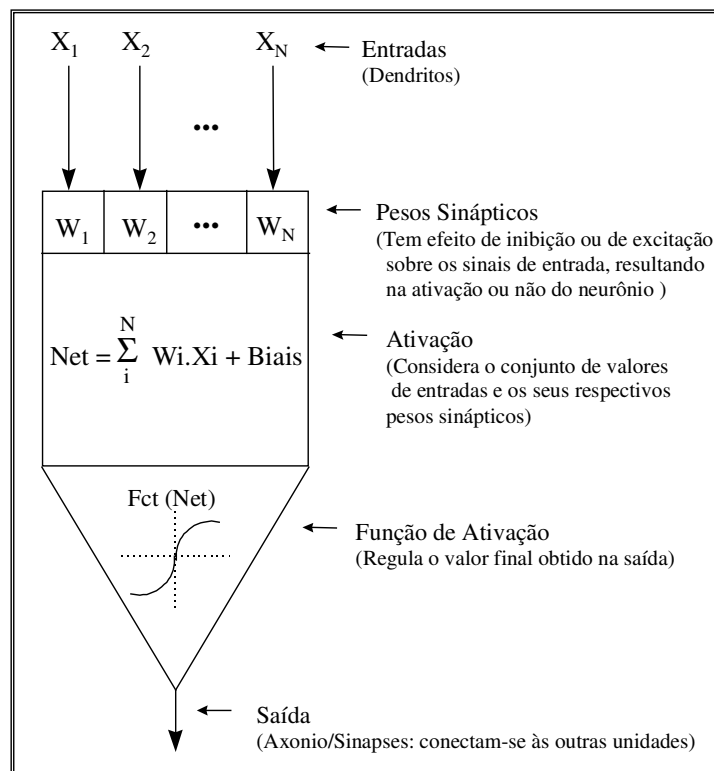


Figura 2.3. Exemplo de Neurônio Artificial

Apesar dos estudos sobre as redes neurais artificiais serem considerados como pertencentes à uma área “jovem”¹ de pesquisa, encontramos atualmente uma série de referências e exemplos de importantes aplicações práticas deste tipo de método de aprendizado, onde podemos citar alguns exemplos de obras relevantes na área, como por exemplo [Fiesler 97, Ripley 96, Bishop 95, Arbib 95, Krose 93, Freeman 92, Simpson 90, Faq 99a].

2.3. Modelos Conexionistas

2.3.1. Definição de uma Rede Neural Artificial

As *redes conexionistas* são formadas por um conjunto de unidades elementares de processamento de informações fortemente conectadas, que denominamos de neurônios artificiais. Uma RNA é constituída por um *grafo orientado e ponderado*. Os *nós* deste grafo são *autômatos simples*, os chamados *neurônios artificiais*, que formam através de suas conexões um autômato mais complexo, a rede neural, também conhecida como rede conexionista.

Cada unidade da rede é dotada de um estado interno, que nós vamos denominar de *estado de ativação*. As unidades podem propagar seu estado de ativação para as outras unidades do grafo, passando pelos arcos ponderados, que nós chamamos de *conexões*, *ligações sinápticas* ou simplesmente de *pesos*

¹ Os primeiros estudos sobre as redes neurais artificiais remontam aos anos 40 (McCulloch-Pitts), mas foi somente na década de 80 que houve um grande desenvolvimento nesta área (Back-Propagation, Hopfield, Kohonen SOFM, ...)

sinápticos. A regra que determina a ativação de um neurônio em função da influência vinda de suas entradas, ponderadas pelos seus respectivos pesos, se chama *regra de ativação* ou *função de ativação*.

As mudanças realizadas nos valores dos pesos sinápticos ou na estrutura de interconexão das unidades de uma rede, são responsáveis pelas alterações no comportamento de ativação desta rede. Estas alterações nas conexões e na estrutura da rede é o que nos permite realizar o *aprendizado* de um novo comportamento. Desta maneira vamos poder modificar o estado de ativação na saída da rede em resposta a uma certa configuração de entradas. Portanto, a rede é capaz de estabelecer associações de entrada-saída (estímulo e resposta) a fim de se adaptar a uma situação proposta. No caso de uma rede com aprendizado supervisionado (vide item sobre tipos de aprendizado), a rede deve adaptar os seus pesos de maneira à passar a responder de acordo com o exemplo dado, ou seja, gerando na sua saída um estado de ativação compatível para com o esperado. O método utilizado para modificar o comportamento de uma rede é denominado de *regra de aprendizado*.

A grande quantidade de modelos de redes conexionistas existentes torna difícil para nós a descrição exaustiva de todos eles. Se o leitor assim desejar, poderá se aprofundar em maiores detalhes sobre os diferentes modelos de RNAs em obras como o “Handbook of Neural Computation” [Fiesler 97]. Nós iremos nos concentrar aqui em diferenciar estes modelos, tomando como base as suas principais características.

2.3.2. Classificação e Propriedades

A grande quantidade de modelos existentes nos leva à uma análise de suas principais propriedades e diferenças em detrimento de uma análise caso à caso mais detalhada. Este estudo das principais propriedades das redes neurais nos permite compreender melhor as vantagens e/ou inconvenientes da escolha de um modelo em detrimento de um outro. Consideramos que não existe apenas uma maneira de classificar todos os modelos, mas de um modo geral devem ser considerados grupos de atributos, tais como: tipo de aprendizado, arquitetura de interconexões, forma interna de representação das informações, tipo de aplicação da rede, etc. Caso o leitor tenha o interesse de buscar uma proposta mais formal de classificação das redes neurais, esta pode ser encontrada em [Fiesler 97].

2.3.2.1. *Aprendizado Conexionista*

O aprendizado conexionista é em geral um processo gradual e iterado, onde os pesos são modificados várias vezes, pouco à pouco, seguindo-se uma regra de aprendizado que estabelece a forma como estes pesos são alterados. O aprendizado é realizado utilizando-se um conjunto de dados de aprendizado disponível (base de exemplos). Cada iteração deste processo gradativo de adaptação dos pesos de uma rede neural, sendo feita uma apresentação completa do conjunto de dados, é chamada de *época de aprendizado*. Os métodos de aprendizado neural podem ser divididos em três grandes classes, segundo o grau de controle dado ao usuário

- **Aprendizado supervisionado:** o usuário dispõe de um comportamento de referência preciso que ele deseja ensinar a rede. Sendo assim, a rede deve ser capaz de medir a diferença entre seu comportamento atual e o comportamento de referência, e então corrigir os pesos de maneira a reduzir este erro (desvio de comportamento em relação aos exemplos de referência). O aprendizado supervisionado utiliza conhecimentos empíricos, habitualmente representados por um conjunto de exemplos etiquetados, ou seja, exemplos com pares de dados de entrada com a respectiva saída associada. A tabela verdade de uma operação booleana do tipo *AND* poderia ser considerada como um conjunto de exemplo de aprendizado, pois indica os valores de entrada e também a saída desejada. Nos casos de problemas de classificação, a saída é a

classe à qual cada exemplo está associado. Exemplo de aplicação: reconhecimento de caracteres em uma aplicação do tipo OCR (*Optical Character Recognition*) [Osório 91].

- **Aprendizado semi-supervisionado:** o usuário possui apenas indicações imprecisas (por exemplo: sucesso/insucesso da rede) sobre o comportamento final desejado. As técnicas de aprendizado semi-supervisionado são chamadas também de aprendizado por reforço (*reinforcement learning*) [Sutton 98]. Para ser mais exato, neste tipo de aprendizado nós dispomos apenas de uma avaliação qualitativa do comportamento do sistema, sem no entanto poder medir quantitativamente o erro (desvio do comportamento em relação ao comportamento de referência desejado). Exemplo: aplicações em robótica autônoma, onde supondo uma situação hipotética, sabemos que seguir em frente não é possível pois existe um obstáculo, mas em compensação não temos uma medida numérica que indique para que lado seguir e exatamente como devemos proceder para desviar deste obstáculo.
- **Aprendizado não-supervisionado:** os pesos da rede são modificados em função de critérios internos, tais como, por exemplo, a repetição de padrões de ativação em paralelo de vários neurônios. O comportamento resultante deste tipo de aprendizado é usualmente comparado com técnicas de análise de dados empregadas na estatística (e.g. *clustering*). Exemplo: diferenciar tomates de laranjas, sem no entanto ter os exemplos com a sua respectiva classe etiquetada (e.g. *self-organizing feature maps* [Kohonen 87]).

O aprendizado conexionista em geral precisa de uma grande quantidade de dados, que nós agrupamos em uma *base de aprendizado*. De acordo com a técnica de aprendizado utilizada, outros conjuntos de dados podem também ser necessários, principalmente para que se possa medir a validade do aprendizado realizado pela rede (e.g. *cross-validation* [Krogh 95]). Este conjunto de dados complementar é usualmente chamado de conjunto de *teste de generalização*. A figura 2.4 apresenta um gráfico típico da evolução do erro durante o aprendizado de uma rede neural, comparando a curva do erro (aprendizado supervisionado) referente à base de aprendizado com a curva do erro da base de teste de generalização.

Nós chamamos de *generalização* a capacidade de um modelo de aprendizado responder corretamente aos exemplos que lhe são apresentados, sendo que estes exemplos *NÃO* devem estar presentes na base de aprendizado. Um modelo que tem uma boa generalização é aquele modelo que responde corretamente aos exemplos contidos na base de aprendizado, mas também a outros exemplos diferentes daqueles da base de aprendizado, e que estão contidos em uma base de teste. A capacidade de generalizar é a principal capacidade buscada nas tarefas que envolvem aprendizado.

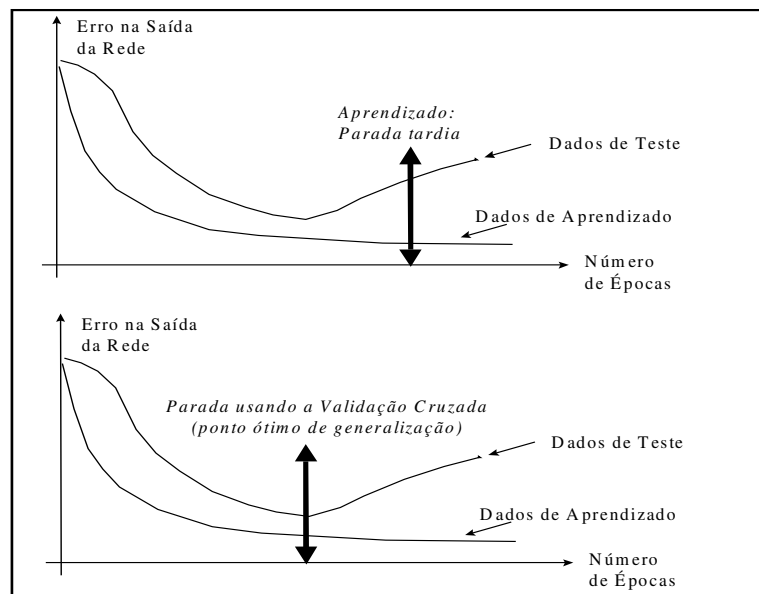


Figura 2.4. Aprendizado: Erro em relação ao conjunto de aprendizagem e de teste

Uma rede pode se especializar demasiadamente em relação aos exemplos contidos na base de aprendizagem. Este tipo de comportamento vai nos levar a um problema de aprendizagem conhecido como super-aprendizado (*over-training / over-fitting*). Normalmente o over-fitting pode ser detectado/evitado através do uso de um teste de generalização por validação cruzada (*cross-validation*).

O aprendizagem de um conjunto de dados pode ser realizado de diferentes formas, se considerarmos a maneira pela qual a rede é alimentada por estes dados:

- **Aprendizado instantâneo:** o conjunto de dados de aprendizagem é analisado uma única vez e com isto o conjunto de pesos da rede é determinado de maneira imediata em uma única passagem da base de exemplos. Este modo de aprendizagem também é conhecido como: *one single epoch learning / one shot learning*.
- **Aprendizado por pacotes:** o conjunto de dados de aprendizagem é apresentado à rede várias vezes, de modo que possamos otimizar a resposta da rede, reduzindo os erros da rede e minimizando o erro obtido na saída desta. Este modo de aprendizagem é caracterizado por trabalhar com uma alteração dos pesos para cada época, ou seja, para cada passagem completa de todos os exemplos base de aprendizagem. O algoritmo de aprendizagem deve reduzir pouco à pouco o erro de saída, o que é feito ao final de cada passagem (análise) da base de exemplos de aprendizagem. Neste tipo de processo, podemos apresentar os exemplos na ordem em que se encontram, ou de modo mais usual, apresentar os dados em uma ordem aleatória. Outros tipos de seleção de exemplos para análise pelo algoritmo de aprendizagem nos levam a métodos como a aprendizagem ativa (vide mais abaixo). Este método é conhecido pelo nome de batch-learning e constitui-se de um dos métodos mais utilizados.
- **Aprendizado contínuo:** o algoritmo de aprendizagem leva em consideração continuamente os exemplos que lhe são repassados. Se o conjunto de dados é bem delimitado, chamamos este método de aprendizagem on-line, e caso o conjunto de dados possa ir aumentando (sendo adicionados novos exemplos no decorrer do tempo), então chamamos este método de aprendizagem incremental. O aprendizagem on-line se opõe ao aprendizagem por pacotes, pois ao contrário deste, para cada novo exemplo analisado já se realiza uma adaptação dos pesos da rede, com o objetivo de convergir na direção da solução do problema. O aprendizagem contínuo

incremental deve ser analisado sob o ponto de vista da aquisição dos dados (adição de novos exemplos na base de aprendizado), onde devemos prestar atenção para não confundir este tipo de aprendizado com o aprendizado incremental em relação a estrutura da rede (adição de novos neurônios no decorrer da simulação). O principal problema do aprendizado contínuo é a dificuldade de achar um bom compromisso entre a plasticidade e a estabilidade da rede. Uma rede com uma grande facilidade de adaptação pode “esquecer” rapidamente os conhecimentos anteriormente adquiridos e uma rede com uma grande estabilidade pode ser incapaz de incorporar novos conhecimentos.

- **Aprendizado ativo:** este modo de aprendizado assume que o algoritmo de adaptação da rede pode passar de uma posição passiva (apenas recebendo os dados do jeito como lhe são passados), para uma posição ativa. Sendo assim, assumimos que este algoritmo poderá vir a intervir sobre a forma como os dados lhe são repassados. Neste caso, a rede pode intervir e determinar assim quais dados que serão considerados e/ou desconsiderados, além também de determinar a ordem em que estes dados deverão ser considerados. A rede pode também vir a solicitar novos dados que julgue necessários para o bom aprendizado do problema proposto. Esta é uma área que vem sendo investigada com mais destaque recentemente.

A adaptação/otimização dos pesos também pode ser implementada por diferentes métodos, segundo o tipo de regra de aprendizado que for empregado. As regras de aprendizado mais usadas são [Jodoin 94, Caudill 92, Simpson 90, Faq 99]:

- Métodos de correção do erro, tais como a descida de uma superfície de erro baseada no gradiente. Exemplos de modelos deste tipo: Adaline, Perceptron, Back-Propagation, Cascade-Correlation;
- Métodos de aprendizado por reforço. Exemplos: Driver-Reinforcement Learning, AHC;
- Métodos de aprendizado por competição ou por auto-organização. Exemplos: Kohonen Self-Organizing Feature Maps, ART1;
- Métodos de aprendizado através da criação de protótipos ou *clusters*. Exemplos: RBF, ART1, ARN2;
- Métodos de aprendizado baseados em memórias associativas (auto-associativas ou hetero-associativas). Exemplos: Modelo de Hopfield, BAM.
- Métodos de aprendizado de seqüências temporais (redes recorrentes). Exemplos: SRN, BPTT, RTRL.

Existem alguns métodos que podem pertencer a duas categorias ao mesmo tempo, por exemplo, as redes com aprendizado do tipo ARN2 [Giacometti 95] que inclui neste modelo técnicas de aprendizado não-supervisionado, aprendizado supervisionado, adaptação por competição, e também através do uso de um método de criação de protótipos. Nas seções seguintes, nós vamos focar com mais atenção os modelos baseados no Perceptron: que possuem aprendizado supervisionado com descida do gradiente.

2.3.2.2. Tipos de Unidades

As unidades de uma rede – os neurônios artificiais – podem ser de diferentes tipos, de acordo com a função interna utilizada para calcular o seu estado de ativação. As principais diferenças são relativas ao

tipo de função de ativação utilizada (e.g. linear, sigmoide assimétrica (exp), sigmoide simétrica (tanh), gaussiana, etc) [Jodoin 94a, Jodoin 94b]. Outro elemento importante diz respeito a forma como os neurônios armazenam as informações: unidades baseadas em protótipos, unidades do tipo Perceptron. Vamos diferenciar aqui estes dois tipos de maneiras de representar o conhecimento nas unidades de uma rede.

- **Redes à base de protótipos:** este tipo de rede utiliza neurônios que servem para representar protótipos dos exemplos aprendidos – as unidades tem uma representação interna que agrupa as características comuns e típicas de um grupo de exemplos [Orsier 95]. As redes baseadas em protótipos tem normalmente um aprendizado não supervisionado (com um ou mais protótipos associados à cada classe). Uma das vantagens deste tipo de redes é a possibilidade de fazer um aprendizado contínuo e incremental, uma vez que não é muito difícil de conceber um algoritmo capaz de aumentar a rede neural através da adição de novos protótipos. Os protótipos são também denominados de clusters, onde apresentamos um exemplo de rede a base de protótipos na figura 2.5. Este tipo de redes vão gerar uma representação dita *localista* de conhecimentos.

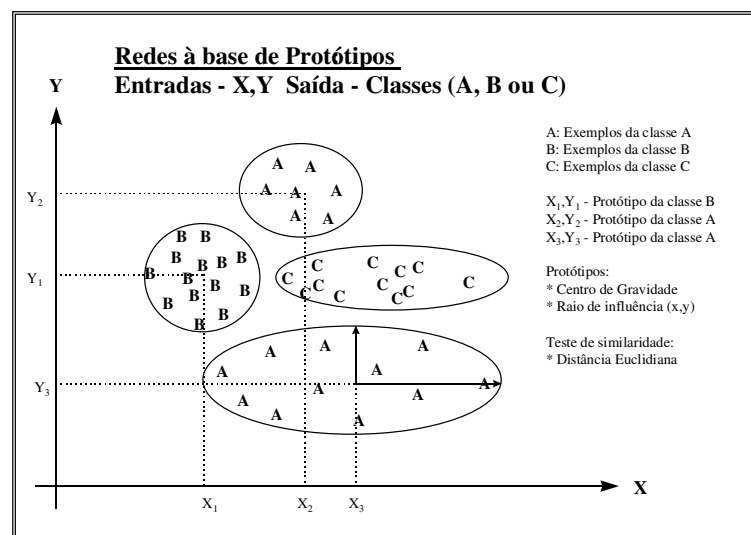


Figura 2.5. Protótipos de uma rede neural com duas entradas

- **Redes à base de Perceptrons:** as unidades do tipo “Perceptron” foram criadas por Frank Rosenblatt em 1950. Este é um dos modelos de neurônios mais utilizados na atualidade. Ele é a base de diversos tipos de RNA com aprendizado supervisionado utilizando uma adaptação por correção de erros (usualmente baseada na descida da superfície de erro usando o gradiente). O modelo do Perceptron de múltiplas camadas (MLP – *Multi-Layer Perceptron*) tornou-se muito conhecido e aplicado, sendo na maior parte das vezes associado a regra de aprendizado do Back-Propagation [Jodoin 94, Widrow 90, Rumelhart 86]. A figura 2.6 apresenta um esquema da representação de conhecimentos nas redes baseadas em Perceptrons, e como este tipo de redes é capaz de classificar padrões, gerando planos (ou hiper-planos) de divisão do espaço em que se situam os exemplos.

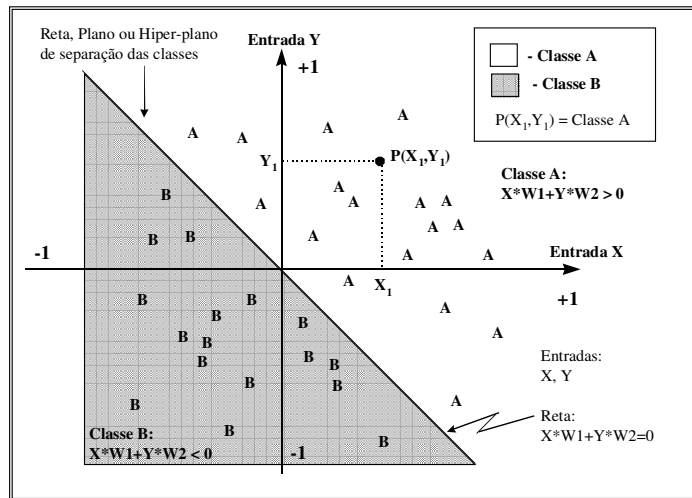


Figura 2.6. Separação de classes (classificação) através do uso de um Perceptron

2.3.2.3. Tipos de Arquiteturas de Conexão das Redes

As unidades de uma rede neural podem se conectar de diferentes modos, resultando em diferentes arquiteturas de interconexão de neurônios. A figura 2.7 apresenta alguns exemplos de possíveis maneiras de conectar os componentes de uma RNA. As arquiteturas de redes mais importantes são:

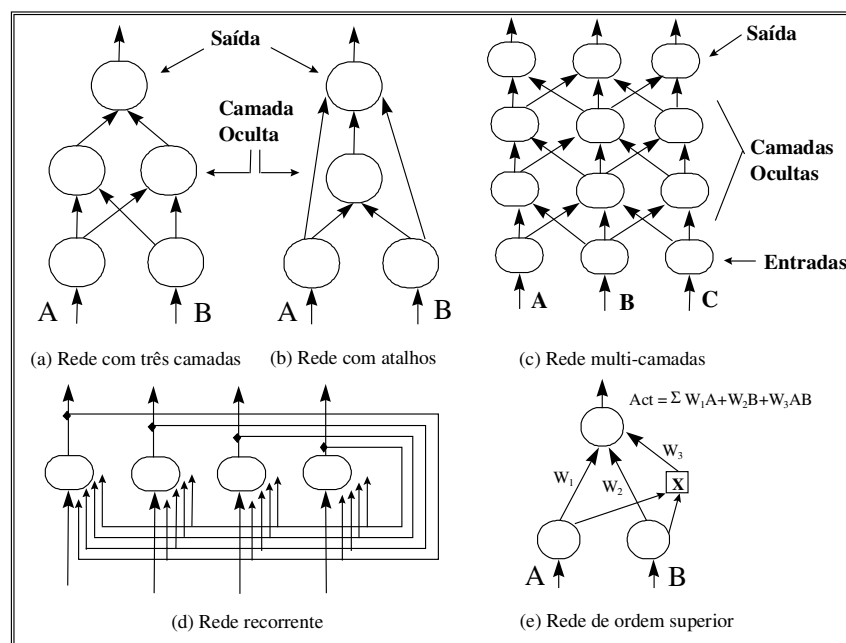


Figura 2.7. Arquiteturas de interconexão de neurônios em redes

- **Redes com uma única camada:** as unidades estão todas em um mesmo nível. Neste tipo de arquitetura, as unidades são conectadas diretamente às entradas externas e estas unidades servem também de saídas finais da rede. As redes de uma única camada possuem normal-

mente conexões laterais (entre os neurônios de uma mesma camada). Um exemplo deste tipo de arquitetura de redes são as redes do tipo “*Self-Organizing Feature Maps*” [Kohonen 87].

- **Redes com camadas uni-direcionais:** as unidades são organizadas em vários níveis bem definidos, que são chamados de camadas ou *layers*. Cada unidade de uma camada recebe suas entradas vindas à partir de uma camada precedente, e envia seus sinais de saídas em direção a camada seguinte. Estas redes são conhecidas como redes *feed-forward*. A Figura 2.7(a) mostra um exemplo de uma rede de três camadas uni-direcionais. Esta arquitetura de três camadas (entrada, camada oculta e saída) é muito usada em aplicações práticas das redes neurais. O modelo MLP [Widrow 90, Rumelhart 86] é composto em geral de uma arquitetura deste tipo, ou seja, com apenas uma camada oculta (*hidden layer*), mas nada nos impede de colocar mais de uma camada oculta entre a camada de entrada e a camada de saída de uma rede. Um outro tipo de interconexão utilizado em redes uni-direcionais são os atalhos (*short-cuts*) que permitem a conexão de uma unidade à outra em uma camada posterior, passando por cima de outras camadas intermediárias. O uso desta técnica vai nos permitir “saltar” por cima de uma camada até uma outra camada (vide figura 2.7(b)), à condição de não introduzir uma recorrência na rede, o que descaracterizaria esta rede como sendo do tipo *feed-forward*.
- **Redes recorrentes:** as redes recorrentes podem ter uma ou mais camadas, mas a sua particularidade reside no fato de que temos conexões que partem da saída de uma unidade em direção a uma outra unidade da mesma camada ou de uma camada anterior à esta. Este tipo de conexões permitem a criação de modelos que levam em consideração aspectos temporais e comportamentos dinâmicos, onde a saída de uma unidade depende de seu estado em um tempo anterior. Os laços internos ao mesmo tempo que dão características interessantes de memória e temporalidade as redes, tornam este tipo de redes muito instáveis, o que nos obriga a usar algoritmos específicos (e usualmente mais complexos) para o aprendizado destas redes. Um tipo particular de redes recorrentes são as redes totalmente conectadas, e um exemplo de modelo recorrente de uma única camada e totalmente conectado são as redes de Hopfield, representadas na figura 2.7(d).
- **Redes de ordem superior:** as unidades deste tipo de rede permitem a conexão direta entre duas ou mais de suas entradas, antes mesmo de aplicar a função de cálculo da ativação da unidade [Fiesler 94a]. Este tipo de rede serve para modelar “sinapses de modulação”, ou seja, quando uma entrada pode modular (agir sobre) o sinal que vem de uma outra entrada. Um modelo particular de rede de ordem superior são as redes tipo *Sigma-Pi* que foram apresentadas no livro PDP – Parallel Distributed Processing [Rumelhart 86], e que são representadas na figura 2.7(e).

A arquitetura de uma rede também pode ser classificada de acordo com a evolução desta no decorrer de sua utilização e desenvolvimento do aprendizado. Em função deste critério podemos ter os seguintes grupos:

- **Redes com estrutura estática:** a rede tem a sua estrutura definida antes do início do aprendizado. A quantidade de neurônios, assim como a sua estrutura de interconexões, não sofrem alterações durante a adaptação da rede. As únicas mudanças se realizam à nível dos pesos sinápticos, que são modificados durante o processo de aprendizado. Este tipo de modelo impõe uma dificuldade maior ao usuário: a determinação do número ideal de neurônios e de conexões a ser utilizado em uma determinada aplicação. Uma rede com poucas unidades e conexões tem forte chance de não ter sucesso em uma tarefa de aprendizado, não tendo condições de alcançar o melhor desempenho possível por falta de capacidade de representação de todos os conhecimentos envolvidos no problema tratado. Uma rede com muitas unidades pode ter também problemas de

convergência e principalmente de generalização, pois quando se tem muita capacidade de armazenamento de informações em uma rede, esta tem uma tendência a decorar os exemplos no lugar de “aprendê-los” (generalizar os conhecimentos sobre o problema) [Fiesler 97, Krogh 95]. No caso deste tipo específico de redes, não existe um método formal que permita determinar o número exato e ótimo de unidades e conexões à serem empregadas no aprendizado de um determinado problema. As redes do tipo MLP com Back-Propagation, de acordo com o modelo proposto por Rumelhart, são redes do tipo estático.

- **Redes com estrutura dinâmica:** as redes que possuem uma estrutura dinâmica são redes onde o número de unidades e conexões pode variar no decorrer do tempo. Estas redes são também chamadas de *ontogênicas* [Fiesler 94b]. As modificações na estrutura da rede podem ser do tipo generativo (incremental) ou do tipo destrutivo (reduzidor por eliminação/simplificação). A escolha entre estes dois tipos de métodos é bastante polêmica: devemos começar com uma rede pequena e ir aumentando ela, ou devemos começar com uma rede bastante grande e ir reduzindo o seu tamanho posteriormente? Alguns autores defendem a idéia de uma criação construtiva de conhecimentos [Elman 93, Osório 98]. Do ponto de vista relacionado à carga de processamento de dados necessária para as simulações neurais, a opção por uma rede pequena que adiciona pouco à pouco novas unidades e conexões é sem dúvida a de melhor performance, pois nas redes do tipo destrutivo uma grande parte do esforço de aprendizado acaba sendo depois destruído ao ser realizada a simplificação da rede. Apesar desta discussão, sobre qual dos dois tipos de redes com estrutura dinâmica que seria melhor usar, não possui um consenso, podemos dizer que uma grande parte dos pesquisadores concorda que as redes ontogênicas em geral são um dos melhores métodos que existem para se escolher uma boa arquitetura para uma rede neural e assim resolver melhor um certo problema proposto. As redes do tipo Cascade-Correlation (CasCor [Fahlman 90]) são redes do tipo dinâmico e incremental.

O último ponto relevante que vamos abordar em relação a arquitetura das redes neurais está relacionado à modularidade [Ronco 96, Ronco 95, Amy 96, Rouzier 98, Jacobs 91]. As redes neurais podem trabalhar com arquiteturas modulares: elas podem ser constituídas por blocos com uma maior ou menor dependência entre eles. Existem diferentes maneiras de integrar e fazer cooperar os diferentes módulos de uma rede neural. Um primeiro método consiste em decompor o problema e obter assim módulos especializados para cada sub-problema. Um exemplo de aplicação deste tipo de método é o caso das aplicações de classificação em múltiplas classes, onde o problema de identificação de cada classe pode ser tratado por módulos separados, e então no lugar de ter um único classificador para os N exemplos em M classes, temos um classificador para cada uma das M classes. Outro tipo de método usado pelas redes modulares, mas mais complexo de ser implementado, é aquele onde os diferentes módulos vão tentar cooperar entre si a fim de juntos resolverem um problema. Neste tipo de método não são impostas tarefas particulares à módulos pré-especificados, deixando para a rede a tarefa de distribuir os conhecimentos e gerenciar a interação entre os módulos.

A modularidade é um problema relativo à escolha de uma arquitetura de rede, mas ela também pode ser ligada ao problema de particionamento dos dados de aprendizado (em um esquema semelhante ao usado na aprendizagem ativa, onde cada módulo poderia escolher que informações iria tratar). Para concluir, devemos salientar que a modularidade pode se tornar um aspecto muito importante a ser considerado segundo o tipo e a complexidade do problema à ser tratado.

2.3.2.4. Tipos de Aplicações das Redes Neurais

As RNA podem ser aplicadas à diferentes tipos de tarefas, tais como: o reconhecimento de padrões (e.g. reconhecimento de faces humanas), a classificação (e.g. reconhecimento de caracteres - OCR), a transformação de dados (e.g. compressão de informações), a predição (e.g. previsão de séries

temporais, como as cotações da bolsa de valores, ou também, uso para o diagnóstico médico), o controle de processos e a aproximação de funções (e.g. aplicações na área da robótica). Todas estas tarefas podem ser reagrupadas em dois grupos principais, segundo o tipo de saída fornecido pela rede neural e o comportamento que é buscado. Estes dois grupos são:

- **Redes para a aproximação de funções:** este tipo de redes devem ter uma saída com valores contínuos e usualmente são empregadas para realizar aproximações de funções (interpolação) [Chentouf 97]. Neste tipo de aplicações, as funções são representada por um conjunto de pontos-exemplo desta. Este tipo de redes é capaz de aprender uma função de transformação (ou de associação) de valores de entrada em valores de saída, usualmente estimando por interpolação as respostas para os casos que não aparecem na base de exemplos. Este tipo de problemas de aprendizado neural de funções é conhecido por ser uma aplicação de um *problema de regressão* [Bishop 97]. Em geral as funções à serem aprendidas pelas redes possuem tanto as entradas como as saídas indicadas através de valores contínuos (variáveis não discretas).
- **Redes para a classificação de padrões:** este tipo de redes deve atribuir para cada exemplo que lhe é fornecido uma classe ao qual este exemplo pertence. Portanto, a saída da rede é a classe associada ao exemplo e por conseqüência, as classes são valores discretos e não contínuos. A classificação é um caso particular da aproximação de funções onde o valor de saída da rede é discretizado e pertence a um conjunto finito de classes. No caso do aprendizado supervisionado, o conjunto de classes é bem definido e conhecido antes de ser iniciado o processo de aprendizado. Uma rede utilizada para fins de classificação deve possuir saídas discretas, ou então, deve implementar métodos de discretização de suas saídas (e.g. aplicação de um limiar de discriminação – *activation threshold*). As entradas da rede podem ser tanto contínuas, como também podem ser discretas, o que não deve interferir no fato desta rede ser usada para uma aplicação classificação.

Seria muita pretensão de nossa parte se tentássemos classificar todos os diferentes modelos de redes neurais em apenas uma destas duas classes descritas acima. A maioria dos modelos pode ser adaptado para ser utilizado em um ou em outro tipo de aplicação, entretanto, alguns modelos são claramente mais adaptados a um tipo de tarefa que ao outro, como é o caso do Cascade-Correlation que foi desenvolvido basicamente apenas para tarefas de classificação.

2.2. Discussão sobre as Redes Conexionistas

As redes conexionistas utilizam métodos de aprendizado à partir de exemplos que possibilitam o ajuste dos pesos de suas conexões, resultando em um comportamento próximo ou até mesmo exatamente igual ao esperado. Esta modificação dos pesos da rede é feita de forma que a rede generalize os conhecimentos contidos na base de exemplos de aprendizado.

Uma boa definição das redes conexionistas é dada por Giacommetti [Giacometti 92]: as redes são capazes de aprender e representar o “saber fazer algo” (*savoir-faire*), que se traduz pelo seu comportamento após o processo de aprendizado de uma tarefa; este “saber fazer”, representado pelos **conhecimentos práticos** (*empirical knowledge ~ practical examples*) adquiridos pela rede, aparece aqui em oposição ao “saber sobre algo” (*savoir-que*) que representa os **conhecimentos teóricos** sobre um determinado assunto (*theoretical knowledge ~ symbolic rules*). Esta distinção entre o saber fazer uma tarefa e o conhecimento sobre a tarefa, é um dos pontos mais importantes de nossa discussão sobre os sistemas híbridos, pois como veremos mais adiante, ambos os conhecimentos se completam um ao outro.

Antes de prosseguir em nossa análise sobre as redes conexionistas, cabe ressaltar que não seria possível fazer aqui uma análise relativa a *todos* os modelos de redes neurais, por isso nosso estudo se concentra principalmente em dois tipos principais de redes: as redes à base de protótipos com aprendizado supervisionado e as redes à base de Perceptrons (MLP) com aprendizado supervisionado. Nosso objetivo é fazer uma discussão orientada principalmente para estes dois tipos de redes, para posteriormente analisar o uso destes modelos junto aos sistemas híbridos inteligentes. Devemos ressaltar também que nosso interesse é voltado às redes usadas para aplicações de classificação, visto que nosso estudo aborda aplicações do tipo sistemas especialistas, onde a aproximação de funções não é uma característica típica deste tipo de sistemas.

As redes conexionistas, em particular aquelas que tem sido aplicadas na construção de sistemas inteligentes, possuem as seguintes vantagens:

- **Conhecimento empírico:** o aprendizado à partir de exemplos é feito de uma maneira bastante simples e permite uma aquisição de conhecimentos de forma automática, muitas vezes de maneira bem mais fácil e confiável do que através de outros métodos de aquisição de conhecimentos (ajuda a resolver em parte o problema do “gargalo da aquisição de conhecimentos” bem conhecido dos sistemas especialistas);
- **Degradação progressiva:** as respostas dadas por uma rede se degradam progressivamente na presença de “perturbações e distorções” dos dados de entradas. Em geral, as redes obtêm uma boa generalização dos conhecimentos presentes na base de aprendizado e sendo assim são menos sensíveis a “perturbações” do que os sistemas simbólicos;
- **Manipulação de dados quantitativos:** o fato de se trabalhar com uma representação numérica dos conhecimentos implica que as redes são melhor adaptadas para a manipulação de dados quantitativos (valores contínuos). Grande parte dos problemas de nosso mundo real, necessitam do tratamento de informações medidas de forma quantitativa, onde uma representação qualitativa muitas vezes implica na perda de informações. As redes neurais são menos vulneráveis aos dados aproximativos e a presença de dados distorcidos ou incorretos que possam estar presentes na base de aprendizado. Esta capacidade de manipular dados aproximados e até mesmo inexatos é mais difícil de ser encontrada em outros métodos de aprendizado do tipo simbólico;
- **Paralelismo em larga escala:** as redes neurais são compostas de um conjunto de unidades de processamento de informações que podem trabalhar em paralelo. Apesar da maioria das implementações de RNAs serem feitas através de simulações em máquinas seqüenciais, é possível de se implementar (softwares e hardwares) que possam explorar esta possibilidade de ativação simultânea das unidades de uma rede. A maior parte das implementações de redes neurais simuladas em máquinas seqüenciais pode ser facilmente adaptada em uma versão paralela deste sistema.

As redes conexionistas apresentam um certo número de inconvenientes, do mesmo modo que os outros tipos de métodos de aprendizado. No caso específico das redes, temos limitações tais como:

- **Arquitetura e parâmetros:** não existe um método totalmente automático para que se possa escolher a melhor arquitetura possível para um problema qualquer. É bastante difícil de se encontrar uma boa topologia de uma rede, assim como os bons parâmetros de regulação do algoritmo de aprendizado. A evolução do processo de aprendizado é bastante influenciada por estes dois elementos: a arquitetura da rede e os parâmetros de regulação do algoritmo. O sucesso da rede depende bastante de uma boa escolha destes elementos, que variam muito de

um problema para outro. Uma simples troca do conjunto de exemplos de uma base de aprendizado pode nos obrigar a reconfigurar toda a rede;

- **Inicialização e codificação:** os algoritmos de aprendizado conexionista são em geral muito dependentes do estado inicial da rede (devido a inicialização aleatória dos pesos) e da codificação dos dados da base de aprendizado. Uma má escolha dos pesos iniciais da rede, do método de codificação dos dados de entrada, ou mesmo, a ordem de apresentação destes dados, pode levar ao bloqueio do processo de aprendizado (e seu conseqüente fracasso), ou então, pode dificultar bastante o processo de convergência da rede na direção de uma boa solução;
- **Caixa preta:** os conhecimentos adquiridos por uma rede estão codificados no conjunto de valores dos pesos sinápticos, assim como pela maneira pela qual estas unidades se conectam. É extremamente difícil para um ser humano conseguir interpretar diretamente estes conhecimentos. As redes conexionistas são “caixas pretas” onde os conhecimentos ficam codificados de tal forma que estes são ininteligíveis para o utilizador ou até mesmo para um especialista. Uma rede não possui a capacidade de explicitar o tipo de raciocínio que lhe levou a obter uma certa resposta, ao contrário dos sistemas baseados em regras, que por sua vez podem facilmente mostrar a seqüência de regras aplicadas na resolução de um problema;
- **Conhecimentos teóricos:** as redes neurais clássicas não permitem que se utilize os conhecimentos teóricos que possam estar disponíveis sobre um determinado problema que estejamos tratando. Como as árvores de decisão, as redes neurais são orientadas para a aquisição de conhecimentos empíricos (baseados em exemplos). Um modo simplista de se aproveitar algum conhecimento teórico pré-existente, consiste em se converter regras em exemplos (“protótipos” representativos destas regras). Entretanto, este tipo de método não nos garante que a rede será capaz de aprender corretamente estes exemplos, sendo assim, não podemos garantir que ao final do aprendizado todos os conhecimentos teóricos disponíveis estarão bem representados internamente na rede.

Estes tópicos listados acima não cobrem exaustivamente todas vantagens e desvantagens das redes conexionistas, mas permitem que se tenha uma idéia das principais características deste tipo de sistemas. Podemos encontrar uma análise complementar a citada acima em outras obras da área, como por exemplo [Orsier 95, Towell 91, Fahlman 88]. No que diz respeito mais especificamente ao aprendizado usando redes neurais baseadas em MLP com Back-Propagation (um dos modelos de redes neurais mais utilizados na atualidade), podemos listar alguns dos pontos inconvenientes deste modelo:

- **Paralisia do aprendizado:** as redes do tipo MLP com Back-Propagation, devido à maneira como o algoritmo ajusta os pesos, tem a tendência à não mais corrigir estes pesos uma vez que a saída das unidades da rede forneçam valores próximos à 0 ou à 1 (isto se deve ao uso da aplicação da sigmoide e de sua derivada pelo algoritmo Back-Propagation). Este comportamento do algoritmo de aprendizado permite dar uma maior estabilidade ao processo de adaptação dos pesos, mas em compensação pode paralisar o aprendizado da rede. Este problema também é conhecido como o “*flat spot problem*”.
- **Instabilidade e esquecimento catastrófico:** de maneira inversa ao problema da paralisia, as redes também podem sofrer de um problema de instabilidade com a conseqüente perda dos conhecimentos anteriormente adquiridos. Uma vez que as redes realizam a minimização do erro de uma maneira não coordenada, ou seja, as unidades competem entre si a fim de reduzir o erro, isto pode levar a uma constante concorrência entre as unidades. Não importa se uma unidade se adaptou a fim de realizar uma pequena, mas importante tarefa, esta unidade vai

continuar sempre tentando alterar os seus pesos a fim de minimizar o erro global ao máximo possível. Este tipo de “comportamento competitivo” pode nos levar à duas situações: (1) as unidades mudam constantemente de “opinião” durante o aprendizado (conhecido como o *moving target problem*), ou, (2) a rede perde grande parte dos conhecimentos já adquiridos ao tentarmos aprender um novo conjunto de exemplos de aprendizado (conhecido como o esquecimento catastrófico). Portanto este problema envolve a busca de um ponto de equilíbrio entre a grande plasticidade (capacidade de se adaptar) e a estabilidade (necessidade de manter as informações) das redes neurais. Redes neurais com uma grande plasticidade estão sujeitas a ficarem alterando indefinidamente os seus pesos, ou então, destruir uma boa configuração de pesos ao tentar adquirir novos conhecimentos.

- **Escolha dos parâmetros do algoritmo de aprendizado e a velocidade de convergência:** na maior parte das aplicações, a velocidade de convergência de uma rede em direção à um mínimo (local ou global) de erro é realizada de maneira muito lenta. As alterações dos pesos da rede devem ser feitas pouco à pouco de modo a garantir que não se “ultrapasse” o ponto ótimo de mínimo da curva de erro. O algoritmo de Back-Propagation é dotado de dois parâmetros – α e β – que controlam respectivamente a velocidade de aprendizado (*learning speed*) e a inércia na descida da curva de erro (*momentum*). Estes dois parâmetros permitem que o processo de adaptação dos pesos da rede seja acelerado ou retardado, mas é preciso que eles sejam ajustados precisamente para que se obtenha bons resultados. Estes dois parâmetros, que devem ser fornecidos pelo usuário, são essenciais para o bom desempenho do processo de aprendizagem. O problema é que não possuímos métodos precisos de estimar estes valores. Além disso, os valores de α e β são bastante dependentes do tipo de aplicação e da base de exemplos de aprendizado utilizada, devendo ser reconfigurados novamente caso o problema tratado seja alterado. Um valor de α ou β que não seja muito bem escolhido pode levar ao fracasso toda a tentativa de se aprender uma base de dados. É por isso que normalmente o aprendizado de uma base de exemplos é feito com o uso de N conjuntos de configurações de parâmetros do algoritmo, para que se possa ter uma melhor chance de encontrar os valores adequados destes parâmetros.

Para concluir sobre as redes MLP, o algoritmo de Back-Propagation não é um algoritmo incremental, nem ao nível da base de exemplos, e muito menos ao nível da estrutura da rede. A arquitetura da rede é estática e conseqüentemente este continua à ser um problema a mais no que se refere ao aprendizado: como fazer para estimar o número ideal de neurônios para uma dada aplicação?

Apesar de todos estes problemas, o algoritmo Back-Propagation ainda é um dos métodos mais usados junto as redes neurais. Alguns pesquisadores, conscientes dos problemas deste algoritmo, propuseram técnicas para resolver ou reduzir estes problemas [Schiffmann 93, 94]. Podemos citar aqui alguns destes métodos aperfeiçoados de aprendizado: o RPROP [Riedmiller 93], o QuickProp [Fahlman 88], o Gradiente Conjugado (Scaled Conjugated Gradient) [Moller 90] e o Cascade-Correlation [Fahlman 90], bem como as técnicas ontogênicas de aprendizado [Fiesler 94].

2.3. Redes Neurais: A busca de uma solução ótima

As redes neurais possuem algumas limitações e problemas que citamos na seção anterior. Vamos listar aqui alguns pontos que devem ser considerados e discutidos no que se refere as redes neurais e a busca de uma solução ou melhoria do aprendizado neural:

- *Aprendizado incremental:* a rede neural deve ser capaz de adquirir novos conhecimentos, sem no entanto destruir os conhecimentos anteriormente adquiridos. A rede neural deve ser

também capaz de adequar a sua estrutura aos requisitos do problema, sendo que esta deveria poder aumentar de tamanho (quantidade de neurônios) à medida que fosse aumentando a complexidade do problema tratado;

- *Estimativa da topologia:* deve-se buscar métodos que permitam ao usuário criar uma topologia de rede adequada para tratar um determinado problema. O ideal seria dotar as redes neurais de mecanismos que possibilitem que esta ajuste automaticamente a sua estrutura em função do problema tratado;
- *Estimativa dos parâmetros de aprendizado:* um bom algoritmo de aprendizado não deve ser muito dependente de parâmetros externos, e idealmente, a rede deveria poder de maneira automática ajustar todos os seus parâmetros para conseguir um resultado ótimo no aprendizado;
- *Introdução de conhecimentos a priori:* as redes neurais devem permitir que *conhecimentos a priori* sobre o problema possam ser inseridos de maneira a facilitar e adiantar o aprendizado de um determinado problema;
- *Instabilidade e velocidade:* um bom algoritmo de aprendizado deve ser o menos instável possível, com ótimas chances de adaptar os pesos da rede e convergir em direção a uma boa solução, minimizando o mais possível o erro na saída da rede. Este algoritmo deve permitir um aprendizado rápido e eficiente;
- *Abrir a “caixa preta”:* devemos buscar uma solução para o problema da falta de mecanismos para analisar os conhecimentos adquiridos pela rede. Devemos ser capazes de representar os conhecimentos adquiridos pela rede em um formato mais compreensível para os seres humanos;
- *Aprendizado ativo:* a rede neural deve ser capaz de dar um retorno sobre o processo de aprendizado, indicando quais os exemplos que devem ser tratados com uma maior prioridade, ou até mesmo, indicando a necessidade de mais exemplos de uma categoria específica para que o problema possa ser corretamente tratado;
- *Tratamento de informações temporais e contexto:* as redes neurais devem ser capazes de considerar o contexto (possuir memória) e assim poder também tratar informações que evoluem no decorrer do tempo, como por exemplo as séries temporais. As redes recorrentes parecem ser um caminho importante a seguir nesta direção, mas ainda restam problemas a serem resolvidos no que se refere a instabilidade e confiabilidade deste tipo de redes.

Estes itens citados acima seguem sendo pesquisados atualmente, e muitas propostas tem sido apresentadas a fim de solucionar (ou minimizar) os problemas ainda enfrentados pelas redes neurais. Apesar de termos problemas relacionados ao aprendizado neural sem serem completamente solucionados, este tipo de técnica tem adquirido uma importância cada vez maior junto à aplicações que necessitem de uma aquisição automática de conhecimentos. As redes neurais superam em muitos casos os demais métodos automáticos de aquisição de conhecimentos.

BIBLIOGRAFIA

[Amy 96] AMY, Bernard. *Recherches et Perspectives dans le Domaine des Réseaux Connexionnistes*. Rapport de Recherche - DRET, Octobre 1996. Web: <http://www-leibniz.imag.fr/RESEAUX/public.html>
Ftp: <ftp://ftp.imag.fr/pub/LEIBNIZ/RESEAUX-D-AUTOMATES/>

[Arbib 95] ARBIB, Michael A. (Editor) *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995.

[Bishop 97] BISHOP, Christopher. *Classification and Regression*. In: Handbook of Neural Computation (section B6.2). E. Fiesler and R. Beale (Eds.) Institute of Physics and Oxford University Press. New York, NY - U.S.A., 1997. Web: <http://www.idiap.ch/publications/fiesler-96.1.bib.abs.html> or http://www.oup-usa.org/acadref/nc_accs.html

[Bishop 95] BISHOP, C.M. *Neural Networks for Pattern Recognition*, Oxford: Oxford University Press. ISBN 0-19-853849-9. 482 pages.1995.

[Caudill 92] CAUDILL, Maureen & BUTLER, Charles. *Understanding Neural Networks - Vol.1: Basic Networks, Vol.2: Advanced Networks*. Bradford Books, MIT Press, 1992.

[Chentouf 97] CHENTOUF, Rachida. *Construction de Réseaux de Neurones Multicouches pour l'Approximation*. Thèse de Doctorat en Sciences Cognitives, Laboratoire TIRF - INPG, Grenoble - France, Mars 1997.

[Elman 93] ELMAN, Jeffrey L. *Learning and Development in Neural Networks: The Importance of Starting Small*. Cognition, 48(1993), pp.71-99. 1993. Web: <http://crl.ucsd.edu/~elman/>
Ftp: <ftp://crl.ucsd.edu/pub/neuralnets/cognition.ps.Z>

[Fahlman 88] FAHLMAN, Scott E. *An Empirical Study of Learning Speed in Back-Propagation Networks*. Carnegie Mellon University - CMU. Computer Science Technical Report CMU-CS-88-162. September 1988. Web: <http://www.cs.cmu.edu/Reports/index.html>

[Fahlman 90] FAHLMAN, S. E.; LEBIERE, C. *The Cascade-Correlation Learning Architecture*. Carnegie Mellon University - CMU, Computer Science Technical Report - CMU-CS-90-100. February 1990. Web: <http://www.cs.cmu.edu/Reports/index.html>
Ftp: <ftp://archive.cis.ohio-state.edu/pub/neuroprose/fahlman.cascor-tr.ps.Z>

[Faq 99] FAQ. *Faq ANN - Comp.ai.neural-nets* 1999.
Web: <http://www.cis.ohio-state.edu/hypertext/faq/usenet/ai-faq/neural-nets/>

[Faq 99a] FAQ. *Faq ANN - Comp.ai.neural-nets FAQ*, Part 4 of 7: Books, data, etc. 1999.
Web: <http://www.cis.ohio-state.edu/hypertext/faq/usenet/ai-faq/neural-nets/part4/faq-doc-1.html>

[Fiesler 94a] FIESLER, E. *Neural Networks Formalization and Classification*. Computer Standard & Interfaces, Special Issue on Neural Networks Standards, John Fulcher (Ed.). V.16, N.3. Elsevier Sciences Publishers, Amsterdam, June, 1994. Web: <http://www.idiap.ch/idiap-networks.html> .
Ftp: <ftp://ftp.idiap.ch/pub/papers/neural/fiesler.formalization.ps.Z>

[Fiesler 94b] FIESLER, Emile. *Comparative Bibliography of Ontogenic Neural Networks*. Proceedings of the International Conference on Artificial Neural Nets - ICANN'94. Sorrento, Italy, May 1994. Web: <http://www.idiap.ch/idiap-networks.html> .
Ftp: <ftp://ftp.idiap.ch/pub/papers/neural/fiesler.ontogenic-summary.ps.Z>

[Fiesler 97] FIESLER, E. & BEALE, R. *Handbook of Neural Computation*. Institute of Physics and Oxford University Press. New York, NY - U.S.A., 1997. Web: <http://www.idiap.ch/publications/fiesler-96.1.bib.abs.html> or http://www.oup-usa.org/acadref/nc_accs.html

[Freeman 92] FREEMAN, James A., SKAPURA, David M.. *Neural Networks: Algorithms, Applications, and Programming Techniques*. 1. ed., Reading: Addison-wesley, 1992. 401 p. il.

[Giacometti 95] GIACOMETTI, Arnaud. *ARN2 - A Prototype-Based Incremental Neural Network*. WNNDA - Workshop on Neural Network Design and Analysis. University of Geneva. January 1995.

[Jacobs 91] JACOBS, R. A., JORDAN, M. I., NOWLAN, S., and HINTON, G. E. (1991). *Adaptive mixtures of local experts*. In *Neural Computation*, (3), 1-12.
Web: <http://www.ai.mit.edu/projects/cbcl/people/jordan/jordan-hp.html>

[Jodouin 94a] JODOUIN, Jean-François. *Les Réseaux de neurones : Principes et définitions*. Editions Hermès, Paris, 1994.

[Jodouin 94b] JODOUIN, Jean-François. *Les Réseaux Neuromimétiques : Modèles et applications*. Editions Hermès, Paris, 1994.

[Kohonen 87] KOHONEN, Teuvo. *Self-Organization and Associative Memory*. Springer-Verlag Series in Information Science. 1987.

[Krogh 95] KROGH, Anders & VEDELSBY, Jesper. *Neural Network Ensembles, Cross Validation and Active Learning* in NIPS - Advances in Neural Information Processing Systems, Vol. 7, pp. 231-238, The MIT Press, 1995), also available in the neuroprose archive as [krogh.ensemble.ps.Z](http://neuroprose.mit.edu/krogh.ensemble.ps.Z).

[Krose 93] KRÖSE, Ben J. & VAN DER SMAGT, Patrick. *An Introduction to Neural Networks*. University of Amsterdam, 1993. Web: <http://www.fwi.uva.nl/research/ias/> ou <ftp://ftp.wins.uva.nl/pub/computer-systems/aut-sys/reports/neuro-intro/>

[Moller 90] MOLLER, Martin F. *A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning*. Technical Report PB-339 - Computer Science Dept., University of Aarhus, Denmark. November 1990.
Ftp: <ftp://archive.cis.ohio-state.edu/pub/neuroprose/moller.conjugate-gradient.ps.Z>

[Orsier 94] ORSIER, Bruno; AMY, Bernard; RIALLE, Vincent & GIACOMETTI, Arnaud. *A study of the hybrid system SYNHESYS*. Workshop ECAI-94 (European Conference on Artificial Intelligence) - Combining Connectionist and Symbolic Processing. Amsterdam, Agosto1994.

[Orsier 95] ORSIER, Bruno. *Etude et Application de Systèmes Hybrides NeuroSymboliques*. Thèse en Informatique, Laboratoire LIFIA-IMAG, UJF - Grenoble, 1995.
Web: <http://www-leibniz.imag.fr/RESEAUX/public.html> .
Ftp: <ftp://ftp.imag.fr/pub/LEIBNIZ/RESEAUX-D-AUTOMATES/orsier.these.ps.gz>

[Osório 91] OSORIO, Fernando Santos. *Um Estudo sobre Reconhecimento Visual de Caracteres através de Redes Neurais*. Dissertação de Mestrado, CPGCC, UFRGS, Porto Alegre - Brasil. Outubro 1991.

[Osório 98] OSORIO, Fernando Santos. *INSS: Un Système Hybride Neuro-Symbolique pour l'Apprentissage Automatique Constructif*. Thèse de Doctorat (Ph.D.) en Informatique. Laboratoire Leibniz – IMAG / INPG. Grenoble, France. 1998.

Web: <http://www-leibniz.imag.fr/RESEAUX/osorio/These/>

[Riedmiller 93] RIEDMILLER, Martin & BRAUN, Heinrich. *A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm*. Proceedings of the IEEE International Conference on Neural Networks. San Francisco - CA - USA. 1993. Web: <http://i11www.ira.uka.de/~riedml/> (or ~neuro)

[Ripley 96] RIPLEY, B.D. *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press, ISBN 0-521-46086-7. 403 pages. 1996.

[Ronco 95] RONCO, Eric & GAWTHROP, Peter J., 1995. *Modular Neural Networks: a state of the art*. Tech. rep CSC-95026. University of Glasgow.
Web: <http://www.ee.usyd.edu.au/~ericr/pub/techrep.html>

[Ronco 96] RONCO, Eric; GOELLE, Henrik & GAWTHROP, Peter J., 1996. *Modular Neural Network and Self-Decomposition*. Tech. Rep CSC-96012. University of Glasgow.
Web: <http://www.ee.usyd.edu.au/~ericr/pub/techrep.html>

[Rouzier 98] ROUZIER, Sophie. *Réseaux Modulaires*. Thèse de Doctorat en Sciences Cognitives, Eq. Réseaux d'Automates - Lab. LEIBNIZ - IMAG, Grenoble - France, 1998.

[Rumelhart 86] RUMELHART, D.; HINTON, G. & WILLIAMS, R. *Learning Internal Representations by Error Propagation*. In : Rumelhart & McClelland: *Parallel Distributed Processing - Explorations in the Microstructure of Cognition - Vol.1: Foundations*. Cambridge: MIT Press, 1986.

[Schiffmann 93] SCHIFFMANN, W.; JOOST, M. & WERNER, R. *Comparison of Optimized Backpropagation Algorithms*. Proceedings of the European Symposium on Artificial Neural Networks, ESANN'93, Brussels, p.97-104, 1993. Web: <http://www.uni-koblenz.de/~schiff/publications.html>

[Schiffmann 94] SCHIFFMANN, W.; JOOST, M. & WERNER, R. *Optimization of the Backpropagation Algorithm for Training Multilayer Perceptrons*. Technical Report, University of Koblenz, Deutschland. September 1995. Web: <http://www.uni-koblenz.de/~schiff/publications.html>

[Simpson 90] SIMPSON, Patrick K. *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations*. Pergamon Press, 1990.

[Sutton 98] SUTTON, Richard S. & BARTO, Andrew G. *Reinforcement Learning: An Introduction*. MIT Press (A Bradford Book), Cambridge, MA, 1998.

[Towell 91] TOWELL, Geoffrey. *Symbolic Knowledge and Neural Networks: Insertion, Refinement and Extraction*. Ph.D. Thesis, Computer Science Dept., University of Wisconsin-Madison, U.S.A. 1991.
Web: <http://www.cs.wisc.edu/~shavlik/uwml.html>
Ftp: <ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/> (towell.thesis.*.ps)

[Widrow 90] WIDROW, Bernard & LEHR, M. *30 Years of Adaptive Neural Networks : Perceptron, Madaline, and Back-Propagation*. Proceedings of the IEEE, New York, Vol.78, N.9, pp.1415-1441. September 1990.

3. Sistemas Híbridos Inteligentes

3.1. Conceitos Básicos

O conceito de sistema híbrido inteligente ou de método híbrido de aquisição de conhecimentos é bastante amplo e pode englobar diferentes tipos de abordagens. De uma maneira mais geral, podemos dizer que todo o sistema que integre dois ou mais métodos diferentes para a solução de um problema é um sistema híbrido. Nós iremos nos concentrar aqui principalmente em relação aos métodos de aquisição automática de conhecimentos (aprendizado automático / *machine learning*), que permitem a integração dos diferentes métodos de processamento de conhecimentos simbólico e conexionista que foram apresentados anteriormente. Nossa idéia não é de traçar uma lista minuciosa de *todas* as possibilidades de combinações entre os diferentes métodos, o que dificilmente seria possível dada a enorme quantidade de variações sobre este tema. Nosso objetivo é o de apresentar uma visão mais geral sobre os tipos de sistemas híbridos inteligentes mais comuns, apresentando também exemplos práticos do que é feito atualmente em termos de métodos e sistemas híbridos.

Uma questão importante que devemos responder é: “quais as principais motivações que nos levariam à utilização deste tipo de sistemas?”. Apesar dos sistemas híbridos serem mais complexos e difíceis de serem implementados, devido ao simples fato de serem compostos por múltiplos módulos e também porque precisamos fazer estes módulos interagirem, mesmos assim estes sistemas tem sido muito estudados e aplicados. Primeiramente vamos nos concentrar nas principais vantagens advindas de sua utilização:

- A integração de duas técnicas complementares permite que uma complete as deficiências da outra de forma a obter um melhor desempenho.
- Ao usarmos diferentes técnicas de aquisição e de representação de conhecimentos ampliamos a capacidade do sistema de adquirir novas informações, e também, fazemos com que este sistema não tenha apenas uma visão parcial, limitada pelas imposições e restrições de um único método, dificultando a obtenção de uma solução para um determinado problema tratado.
- Algumas tarefas muito complexas que usualmente não podem ser abordadas através da utilização de uma única técnica para a sua solução, podem ser quebradas em sub-problemas, e estes sim podem ser tratados individualmente, chegando-se a uma solução global do problema.
- O processamento em paralelo de informações pelos diferentes módulos irá usualmente nos permitir que tenhamos sistemas com uma maior performance, mais robustos, e com uma maior tolerância a falhas.

Em relação aos sistemas híbridos inteligentes, podemos considerar também que estes, em seu objetivo maior que é a reprodução do comportamento inteligente humano, se caracterizam por:

- Os seres humanos não empregam um único método de solução de problemas, usando diferentes técnicas, como por exemplo, soluções baseadas: em experiências passadas; em conhecimentos teóricos adquiridos; em diferentes métodos de raciocínio – por inferência, por probabilidades, por dedução, por analogia; etc. Assim como os seres humanos, os sistemas inteligentes procuram através da integração de diferentes métodos automáticos “imitar” a inteligência humana.

- Integrar diferentes tipos de representação de conhecimentos: exemplos (casos típicos), contra-exemplos (casos atípicos), regras e predicados lógicos, regras difusas (*fuzzy*), probabilidades, etc.
- Trocar informações e conhecimentos entre os seus módulos, o que usualmente, lhe confere uma característica de um sistema que evolui no tempo, construindo o conhecimento através de interações entre o usuário e o sistema e entre os diferentes módulos do sistema.
- Muitas vezes, ao combinar múltiplos módulos, teremos assim um sistema que deve ser mais robusto, e que aceite certas contradições e “conflitos cognitivos” entre os seus diferentes módulos. Isto até pode vir a ser um problema, mas se for bem explorado poderá nos trazer grandes vantagens.

De maneira bastante informal, podemos concluir esta introdução aos sistemas híbridos citando aqui um ditado popular que diz “que várias cabeças pensam melhor do que uma única”, e adicionando uma pequena observação no que diz respeito ao comportamento humano, que felizmente, difere de uma pessoa para a outra. Portanto o que buscamos com o uso dos sistemas híbridos é a integração de múltiplos módulos que cooperam entre si, cada um com suas especificidades, visando em conjunto o tratamento de um objetivo comum.

3.2. Tipos de Integração em Sistemas Inteligentes

Vamos analisar aqui algumas possibilidades de integração de diferentes tipos de métodos e sistemas de aquisição e representação de conhecimentos empregados nos sistemas inteligentes. Nas seções anteriores, vimos alguns métodos automáticos de aquisição de conhecimento que se destacaram:

- Árvores de decisão (IDT - *Induction of Decision Trees*);
- Sistemas baseados em conhecimentos (KBS - *Knowledge Based Systems*), sistemas especialistas (Expert Systems) e sistemas baseados em regras (*Rule Based/Production Systems*);
- Sistemas baseados em regras de lógica difusa ou nebulosa (*Fuzzy Logic rules*);
- Sistemas baseados em conhecimentos
- Sistemas de aquisição de conhecimentos do tipo ILP (*Inductive Logic Programming*);
- Sistemas baseados em casos/exemplos (CBR – *Case Based Reasoning*);
- Algoritmos Genéticos (GA – *Genetic Algorithms*);
- Redes Neurais Artificiais (ANN – *Artificial Neural Networks*);

Estes métodos tem sido usados de forma combinada, dando origem a múltiplos sistemas híbridos que estão relacionados na bibliografia específica desta área de estudos [Osorio 99]. Vamos tentar dar uma idéia de algumas destas possíveis combinações entre os referidos métodos citados acima.

3.2.1. Sistemas Simbólico-Difusos:

Os sistemas simbólico-difusos integram a lógica nebulosa (*fuzzy logic* [Faq 99b]) e os sistemas especialistas baseados em conhecimentos e/ou regras (sistemas simbólicos “tradicionais”). Este tipo de sistemas é na realidade uma extensão dos sistemas simbólicos, nos quais adicionamos a possibilidade de se representar e manipular regras difusas (*fuzzy rules*). Com isso estes sistemas passam a trabalhar não

apenas com regras que manipulam proposições lógicas, baseadas em valores binários do tipo verdadeiro/falso, passando a manipular também valores numéricos e intervalos (*fuzzy sets*). A principal vantagem deste tipo de sistemas híbridos é a extensão da capacidade de representação e de manipulação de conhecimentos, que podem assim incluir a incerteza associada à lógica nebulosa, gerando sistemas mais robustos.

Um exemplo deste tipo de sistemas é o Fuzzy-CLIPS [ITT 96], uma versão adaptada criada à partir da linguagem CLIPS – C Language Integrated Production System [Giarratano 93]. Um outro sistema bastante conhecido que se utiliza deste tipo de técnicas é o FLIE [Vestli 93]. Podemos mesmo questionar se estes sistemas são realmente híbridos, uma vez que em muitos casos acabam sendo apenas uma extensão de um sistema especialista que incorpora ao seu mecanismo de inferência o tratamento de regras nebulosas.

3.2.2. Sistemas Simbólico-Genéticos:

Estes sistemas são normalmente compostos de um módulo genético responsável pela aquisição de conhecimentos à partir dos dados disponíveis sobre um problema (exemplos de aprendizado), e de um módulo simbólico responsável pela implementação do mecanismo de inferência simbólica (*inference engine*). O algoritmo genético vai fazer uma busca evolutiva à partir de uma base de regras, visando otimizar estas regras em relação a uma base de exemplos disponível. A base final de regras obtidas é então manipulada por um módulo simbólico do tipo convencional.

Podemos citar como exemplos deste tipo de sistemas, as seguintes implementações: DELVAUX [Eick 96], COGIN [Greene 93] e GABIL [DeJong 93]. Alguns autores utilizam o nome “programação genética” para denominar o emprego de algoritmos genéticos que geram automaticamente programas, representados por um conjunto de regras, ou mais usualmente, resultando em programas em LISP.

3.2.3. Sistemas Neuro-Genéticos:

A maior parte dos sistemas neuro-genéticos são concebidos a fim de contornar os problemas relativos à escolha de uma arquitetura de rede neural adequada para um determinado problema, bem como para estimar (otimizar a escolha) os parâmetros de configuração do algoritmo de aprendizado utilizado pela rede neural. As redes que se utilizam de algoritmos genéticos para modificar a sua estrutura, são denominadas de redes evolutivas (evolutionary ANN) [Balakrishnan 95, Nolfi 94, Patel 97, Porto 97].

Dentro do conjunto de sistemas neuro-genéticos que utilizam os algoritmos genéticos para adaptar os parâmetros da rede, podemos destacar um sistema que utilizou esta técnica para acelerar o aprendizado, adaptando os parâmetro de velocidade de aprendizado e da inércia ou *momentum* (α e β) [Salomon 96]. Existem inclusive sistemas que utilizam o algoritmo genético como processo de otimização dos pesos sinápticos, substituindo em parte o algoritmo de aprendizado na adaptação dos pesos da rede [Potter 92].

É importante salientar que os algoritmos genéticos são algoritmos de otimização, que utilizam um método baseado na recombinação e seleção de populações. Sendo assim, podemos considerar então populações de redes neurais, populações de parâmetros de configuração de uma rede, ou mesmo populações de peso. Este tipo de método exige usualmente uma grande quantidade de processamento de dados, visto que se considerarmos uma população de redes neurais com diferentes topologias, isto significa que teremos que testar (simular) cada elemento da população, avaliar seu desempenho, e depois modificá-lo para então recomeçar o processo de avaliação e seleção. Em termos computacionais, este tipo de processamento pode ser muito pesado, mas tende a ser mais eficiente (desde que a função de mutação e seleção sejam bem definidas) do que se este mesmos processo fosse realizado de maneira manual na base da “tentativa e erro”.

3.2.4. Sistemas Neuro-CBR:

A integração de uma rede conexionista e de um sistema baseado em casos permite que se faça generalizações sobre um conjunto de exemplos de um problema, mas sem no entanto deixar de preservar certas particularidades (casos específicos) relacionados a este problema. Um exemplo deste tipo de técnica híbrida de aquisição de conhecimentos é o sistema *ProBis* [Malek 95, 96, 96a]. Este sistema permite a obtenção de uma rede com protótipos (generalização) dos casos aprendidos, e também permite que seja criada uma coleção de casos atípicos (casos particulares) e de casos “fronteira” (localizado entre dois protótipos ou nas extremidades destes), que são armazenados em na memória de casos do sistema CBR.

Os sistemas neuro-CBR nos permitem então buscar um compromisso intermediário entre a generalização total e a especialização individual em relação a uma base de exemplos. Esta proposta vem de encontro à uma das principais idéias de suporte do uso de sistemas híbridos: a complementaridade entre os métodos que compõem o sistema, e que permite que ambos ajudem a suprir as deficiências de um método em relação ao outro. Os sistemas neuro-CBR, de acordo com a sua forma de implementação podem vir a ser classificados dentro da categoria de “sistemas neuro-simbólicos”, que descreveremos no item seguinte.

3.2.5. Sistemas Neuro-Simbólicos:

Este tipo de sistemas híbridos é sem dúvida alguma o mais estudado entre todos os métodos híbridos que foram descritos neste capítulo. Nesta categoria estão todos os sistemas híbridos que combinam a representação de conhecimentos em uma forma simbólica (representação de alto nível, compreensível para o ser humano) e a representação de conhecimentos em uma forma conexionista, ou como também é conhecida, representação sub-simbólica (representação de baixo nível, de compreensão extremamente difícil para um ser humano). Podemos subdividir esta categoria dos sistemas neuro-simbólicos nos seguintes subgrupos:

3.2.5.1. *Sistemas Neuro-Fuzzy:*

Os sistemas híbridos neuro-fuzzy são uma das categorias de sistemas híbridos que mais se desenvolveram, pois a lógica nebulosa e as redes conexionistas possuem muitos pontos em comum. Podemos encontrar na literatura um número significativo de referências sobre este tipo de sistemas [Machado 92, Jang 96, Krzysztof 97, Kasabov 95, Gonzalez 96, Nauuck 95, Hayashi 94, Alche 93, Cox 92, Posey 92, Rocha 92]. Os sistemas neuro-fuzzy são divididos basicamente em três grupos principais:

- Sistemas que integram regras nebulosas dentro das redes neurais, onde o modelo neural do tipo RBF é um dos mais usados neste tipo de integração. Neste tipo de sistemas, um conjunto de regras nebulosas dão origem a uma rede neural (processo chamado de compilação de regras em redes neurais), que pode vir a substituir o mecanismo de inferência usado em conjunto com estas regras nebulosas;
- Sistemas que fazem a extração de regras nebulosas à partir das redes neurais. Estes sistemas permitem extrair automaticamente os conhecimentos adquiridos por uma rede neural, representando estes conhecimentos sob a forma de regras da lógica difusa. Um exemplo deste tipo de método de extração é implementado pelo sistema Fagnis [Cechin 98].
- Sistemas que implementam neurônios com processamento difuso (fuzzy-neurons). Cada neurônio da rede pode ser transformado em um elemento de processamento capaz de realizar operações da lógica difusa.

Podemos citar aqui uma lista de alguns dos principais sistemas neuro-fuzzy (no entanto esta lista resta bastante simplista e incompleta) citados na literatura sobre este tipo de métodos híbridos: FuzzyARTMAP [Carpenter 92], FUN [Sulzberger 93], FuzzyCOPE [Kilgour 97, Kasabov 96], NEFCON-NEFCLASS-NEFPROX [Nauck 97], ANFIS [Jang 93] e FUNEGEN [Halgamuge 94].

3.2.5.2. *Sistemas Neuro-IDT:*

Os sistemas híbridos que combinam redes neurais com as árvores de decisão (IDT) podem ser classificados em três grupos principais:

- Sistemas onde a construção e inicialização das redes é feita à partir de uma árvore de decisão [Lee 95, Sahami 95]. A inserção de uma árvore de decisão “dentro” de uma rede neural nos permite simplificar o problema da escolha de uma topologia e pesos iniciais para uma rede;
- Sistemas onde é feita a extração de conhecimentos da rede neural após o aprendizado, permitindo assim a obtenção de uma árvore de decisão [Boz 97b, Craven 96 e 96b, Medsker 95]. Este problema tenta resolver o problema da dificuldade de compreensão dos conhecimentos adquiridos por uma rede neural. A extração de uma árvore de decisão nos permite que obtenhamos explicações mais concretas sobre a forma como chegamos a uma certa solução para um problema proposto. Toda árvore de decisão pode ser re-escrita sob a forma de um conjunto de regras, logo, ao representarmos o conhecimento da rede sob a forma de uma árvore de decisão, estaremos abrindo esta “caixa preta” que é uma rede conexionista;
- Sistemas neuro-IDT realmente mistos, onde cada nodo da árvore de decisão é substituído por uma unidade de uma rede conexionista. Neste caso, ao invés de termos uma regra de seleção nos nodos da árvore, teremos neurônios que são treinados para realizar uma função equivalente a esta seleção [Alche 93, Utgoff 88].

Este último tipo de sistema neuro-IDT apresenta algumas características interessantes do ponto de vista da combinação dos dois métodos de base que o compõem, visto que a capacidade de escolha e representação de uma decisão contida em um nodo de uma árvore é um fator importante que pode limitar bastante a capacidade do sistema no tratamento de um problema mais complexo.

3.2.5.3. *Sistemas Neuro-KBS:*

Os sistemas que integram as redes conexionistas aos sistemas baseados em conhecimentos (KBS) são usualmente chamados de **Sistemas Híbridos Neuro-Simbólicos** (SHNS), ou também, de **Sistemas Híbridos Simbóli-Conexionistas**. Estes sistemas permitem a integração de um sistema baseado em regras simbólicas (conhecimentos simbólicos, em geral) juntamente com um sistema baseado em uma rede conexionista (conhecimentos sub-simbólicos), onde é importante salientar a complementaridade existente entre estas duas formas de representação e manipulação de conhecimentos.

O número de publicações recentes ligadas a este tema é bastante impressionante, onde podemos citar uma lista extensa, composta de obras importantes desta área, com referências sobre este tipo de sistemas híbridos elaborada por F. Osório [Osorio 99].

Em relação as implementações de sistemas do tipo SHNS, podemos citar alguns dos mais conhecidos sistemas deste tipo:

- Sistema SYNHESYS: KBS + RNA com protótipos [Giacometti 92, Orsier 95];

- Sistema KBANN: Regras simbólicas + RNA tipo MLP [Towell 91];
- Sistema INSS: KBS + RNA tipo MLP incremental [Osório 98, Osório 99a];
- Redes KBCNN: Regras simbólicas + RNA tipo MLP [Fu 94];
- Sistema SCANDAL: KBS + RNA tipo MLP [Stückelberg 96];
- Sistema RAPTURE: Regras + RNA tipo CFBP (probabilista) [Mahoney 96, 93];
- Sistema MACIE: Sistema especialista baseado em redes MLP [Galant 93, 88];
- Sistema RUBICON: Codificação de base de regras em uma rede sem aprendizado [Samad 92];
- Sistema CONSYDERR: Processamento baseado em regras e similaridades [Sun 95a];
- Sistema CORE: Interação entre dois módulos – KBS + RNA [Kasabov 90].

Neste trabalho foi feita uma opção pelo estudo mais aprofundado sobre este tipo de sistemas híbridos, os *sistemas híbridos neuro-simbólicos*, entre os diversos tipos de sistemas híbridos que descrevemos anteriormente. O principal motivo desta escolha é que optamos por uma forma de integração que privilegie a exploração da complementaridade entre os métodos simbólicos e conexionista, dadas as diferentes maneiras como estes representam e manipulam os conhecimentos adquiridos. Nos itens seguintes deste trabalho, vamos nos concentrar de uma maneira quase exclusiva sobre este tipo específico de integração dos SHNS.

3.3. Sistemas Híbridos Neuro-Simbólicos

O uso dos sistemas híbridos, independentemente do tipo de opção feita, busca explorar as vantagens particulares de cada método a fim de vencer as dificuldades e limitações características de cada um destes. Além desta característica, a nossa opção pelos sistemas híbridos neuro-simbólicos se deve também ao fato de buscarmos alcançar os seguintes objetivos descritos abaixo:

- » Possibilidade de explorar todos os conhecimentos disponíveis sobre o problema à ser tratado, ou seja, desejamos explorar tanto os conhecimentos teóricos, quanto os conhecimentos empíricos (exemplos práticos). O sistema de aquisição automática de conhecimentos deve ser capaz de tratar tanto bases de regras simbólicas (conhecimento de alto nível, já assimilado e tratado) bem como bases de exemplos (conhecimento de baixo nível, na sua forma simples pouco tratada);
- » Possibilidade de abordar problemas onde os conhecimentos disponíveis não são nem completos, nem perfeitamente exatos/corretos. Os conhecimentos disponíveis sobre um problema são, na maioria das vezes, conhecimentos imperfeitos que irão evoluir e serem aperfeiçoados (pois usualmente não contêm verdades absolutas) no decorrer do passar do tempo;
- » Possibilidade de tratar e explorar os dados simbólicos de maneira conveniente, assim como os dados numéricos. O uso de conhecimentos representados por informações que contêm variáveis qualitativas e bem como variáveis quantitativas, não deve em

hipótese alguma prejudicar o desempenho do sistema, ou limitá-lo de alguma maneira. Um sistema inteligente deve ser capaz de manipular dados vindos do “mundo real” compostos por valores contínuos, usualmente inexatos e/ou aproximados;

- » Possibilidade de representar os conhecimentos de baixo nível e de alto nível. Os conhecimentos devem ser representados de tal maneira que possamos trabalhar tanto ao nível simbólico quanto ao nível sub-simbólico. Por exemplo: para tratarmos um problema da robótica autônoma, devemos ser capazes de trabalhar tanto com informações ao novo sensorial-motor (sub-simbólico = baixo nível), como ao nível de planificação de tarefas e trajetórias (simbólico = alto-nível). O formalismo de representação de conhecimentos deve ser robusto o bastante para que possamos codificar diferentes tipos de relações e associações entre as informações disponíveis.

Concluindo sobre estes requisitos, podemos afirmar que em particular os sistemas híbridos neuro-simbólicos possuem características que vem de encontro com as propriedades que se deseja obter em um sistema híbrido. A complementaridade entre estes dois métodos permite suprir as deficiências encontradas nos métodos tradicionais quando confrontados com problemas como os citados acima. A tabela que segue abaixo (baseada em dados extraídos de Lallement 96, Boz 95, Osorio 95, Giacometti 92 e Towell 91) realça justamente esta complementaridade entre os modelos simbólico e o conexionista:

I.A. Simbólica	I.A. Conexionista
☺ - Inserção de conhecimentos teóricos sobre o problema pode ser feita de uma maneira direta. Para isso, basta explicitar os conhecimentos do expert e convertê-los em um formalismo de representação de conhecimentos adequado ao sistema.	☹ - Normalmente não é possível que se aproveite diretamente os conhecimentos teóricos disponíveis sobre um problema tratado. É necessário que passemos por uma fase de aprendizado à partir de exemplos, para que possamos adquirir novos conhecimentos.
☹ - O processamento das informações é sequencial, o tempo de resposta do sistema em uma consulta normalmente é longo e cresce consideravelmente com o aumento da base de conhecimentos.	☺ - As redes neurais são compostas de uma série de unidade de processamento de informações que podem operar em paralelo, com um tempo de resposta muito rápido (na fase de consulta aos conhecimentos adquiridos).
☺ - A inserção de conhecimentos (e.g. regras) pode ser feita muito fácil e rapidamente uma vez que estas regras já foram tratadas pelos especialistas e/ou engenheiros de conhecimento.	☹ - O processo de aprendizado pode ser muito longo pois a adaptação dos pesos da rede se dá por um processo de otimização dos pesos, que são alterados pouco à pouco de maneira à manter estável o processo de otimização.
☹ - O aprendizado não é um processo fundamental de sustentação deste tipo de sistemas, que inclusive só incluíram processos de aquisição automática de conhecimentos mais recentemente. A aquisição normalmente é feita	☺ - O aprendizado e a generalização de conhecimentos à partir do tratamento de um conjunto de exemplos são os pontos fortes dos métodos conexionistas. O processo de aprendizado é a base deste tipo de método,

<p>por explicitação de regras por parte do especialista, e em consequência disto temos o famoso problema dos sistemas especialistas que é o “gargalo da aquisição de conhecimentos”.</p>	<p>sendo a principal justificativa de sua existência. As redes neurais não dependem tanto do especialista, e portando podem automatizar o processo de aquisição de conhecimentos, ajudando a contornar o problema do “gargalo da aquisição de conhecimentos”.</p>
<p>☺ - Os métodos simbólicos permitem que sejam obtidas explicações sobre as respostas fornecidas pelo sistema. Através do uso destes métodos podemos justificar as respostas fornecidas, baseando-se no mecanismo de resolução do problema empregado e nos conhecimentos de alto nível codificados na base de conhecimentos deste. Os conhecimentos do sistema são codificados em uma linguagem próxima da linguagem natural, e portando mais facilmente interpretáveis.</p>	<p>☹ - As redes são “caixas pretas” onde os conhecimentos estão codificados em um conjunto de pesos e de interconexões entre os neurônios. Nós não temos acesso a uma forma “mais compreensível” destes conhecimentos que foram adquiridos pela rede. Portanto, é muito difícil para um ser humano que este possa interpretar diretamente os conhecimentos de uma rede, e por consequência fica difícil de justificar as suas respostas.</p>
<p>☹ - Normalmente, os sistemas simbólicos pressupõem o uso de conhecimentos teóricos que sejam ao mesmo tempo corretos e completos sobre o problema tratado. A maior parte dos sistemas simbólicos não é adaptada ao tratamento de informações aproximadas, incompletas ou expressas através de variáveis numéricas contínuas.</p>	<p>☺ - As respostas do sistema se degradam lenta e progressivamente em presença de uma entrada com distorções. As redes são muito bem adaptadas ao tratamento de informações aproximadas, incompletas ou expressas através de variáveis numéricas contínuas.</p>
<p>☺ - Os conhecimentos são representados por regras e por estruturas de dados. Os conhecimentos são ditos de “alto nível”.</p>	<p>☺ - Os conhecimentos são codificados nas redes e representam relações de baixo nível estabelecidas entre as entradas das redes.</p>
<p>☹ - O desenvolvimento de um sistema especialista é uma tarefa árdua e longa. Os sistemas especialistas devem ser bem estudados e estruturados.</p>	<p>☹ - A especificação da arquitetura e dos parâmetros de aprendizado da rede podem se constituir em uma tarefa árdua e longa, onde nem sempre a rede funciona bem.</p>

Tabela 3.1. Propriedades do modelo simbólico e do modelo conexionista

Esperamos que esta tabela tenha servido para demonstrar claramente a complementaridade entre os módulos simbólico e neural, e que ela também tenha servido para justificar este nosso grande interesse em nos voltarmos para os SHNS.

3.3.1. Tipos de Integração e Classificação dos SHNS

Existem vários tipos de classificação dos modos de integração dos sistemas híbridos neuro-simbólicos [Lallement 96, Hilario 96, Orisier 95, Medsker 95]. Estas classificações são mais ou menos parecidas, onde cada autor tenta colocar em evidência um determinado aspecto em relação aos demais. Em vez de tentar criar uma classificação hierárquica e global dos SHNS, nós preferimos considerar um

aspecto de cada vez, fazendo com que um mesmo sistema possa ser classificado em várias categorias ao mesmo tempo. Os critérios de classificação são os seguintes:

- *Tipo de integração*: método unificado, semi-híbrido ou híbrido;
- *Modo de integração*: processamento em cadeia, sub-processamento, meta-processamento, ou co-processamento;
- *Transferência de conhecimentos*: do módulo simbólico para o conexionista, ou do módulo conexionista para o simbólico, ou nos dois sentidos;
- *Grau de acoplamento dos módulos*: acoplamento fraco, acoplamento médio e acoplamento forte;
- *Tipo de representação dos conhecimentos*: localista, distribuída ou combinada;
- *Tipo e poder de codificação dos conhecimentos*: proposições lógicas de ordem 0, proposições de ordem 0⁺ (incluem a manipulação de intervalos de valores), predicados, regras nebulosas, regras probabilistas, árvores de decisão, redes tipo MLP (Perceptron com múltiplas camadas), redes tipo protótipos, etc;
- *Tipo de mecanismo de inferência e de resolução do problema*: “forward chain”, “backward chain”, propagação de ativações neurais, por analogia, por unificação, etc;
- *Modo de aquisição de conhecimentos*: aprendizado contínuo, aprendizado não contínuo, aprendizado supervisionado, aprendizado não supervisionado.

Uma boa parte das classificações apresentadas aqui tem a sua origem em trabalhos desenvolvidos dentro do projeto Europeu MIX [Hilario 93, 94, 96], destacando-se o trabalho dos seguintes pesquisadores: B. Orsier, Y. Lallement e M. Hilário. Nós vamos detalhar aqui os três primeiros grupos, onde o leitor poderá encontrar mais informações sobre os demais grupos nas obras referenciadas acima, ou em [Osorio 98].

3.3.1.1. Tipos de Integração:

Os tipos de integração dos sistemas híbridos neuro-simbólicos (ver a Figura 3.1) são os seguintes:

1. **Método Unificado**: Este tipo de método é conhecido como um método “puro” ou como uma hibridização no sentido amplo. A idéia de base deste tipo de integração é a de que podemos integrar todas as funcionalidades do nível simbólico e conexionista em um único módulo, seja este unicamente conexionista ou unicamente simbólico. Este método unificado se propõe à integrar as propriedades dos sistemas simbólicos + conexionistas em uma única solução unificada, através da ampliação/extensão das capacidades de um dos dois módulos (aquele que for escolhido para unificar tudo). Os sistemas puramente conexionistas tentam implementar através de uma rede neural os processos que seriam realizados por um método simbólico, ou seja, a rede neural é usada para implementar mecanismos de inferência simbólica, de unificação, etc. Estes sistemas puramente conexionistas também são chamados de CSP – Connectionist Symbolic Processing. Os sistemas puramente simbólicos tentam implementar extensões ao processo de inferência simbólica de maneira a permitir a manipulação de incertezas, de valores contínuos e aproximados, de probabilidades, etc.

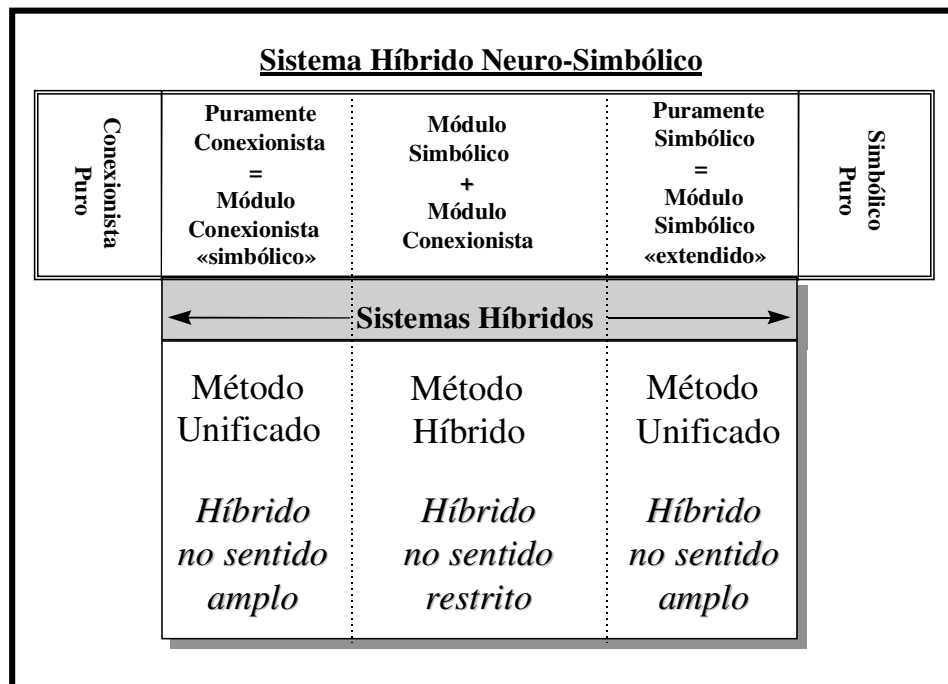


Figura 3.1. Tipos de integração Neuro-Simbólica

2. **Método Semi-Híbrido:** Estes são os métodos onde encontramos os trabalhos relativos a traduções de conhecimentos de um tipo de modelo para outro, sem necessariamente ter uma co-existência e cooperação de ambos na solução dos problemas. Neste grupo aparece a compilação de bases de regras dentro de redes neurais (inserção de conhecimentos) e a explicitação de regras à partir de redes neurais (extração de conhecimentos). Justamente devido ao fato de serem apenas semi-híbridos, muitos destes sistemas possuem implementado apenas um tipo de transferência de conhecimentos, ou a compilação de regras, ou então a extração de regras. Para nós, vamos considerar justamente que à partir do momento que o sistema oferece a possibilidade de inserir e também extrair regras (transferência bi-direcional) este sistema passará a ser considerado como um método híbrido completo (hibridização no sentido amplo). Nós vamos retomar esta discussão mais abaixo, no que se refere as transferências de conhecimentos.
3. **Método Híbrido:** Este tipo de método é baseado em uma verdadeira integração de no mínimo dois módulos, um módulo simbólico e um módulo conexionista. Entretanto nada nos impede de realizar uma integração de múltiplos módulos simbólicos e conexionistas. Em resumo, neste tipo de método iremos encontrar um mecanismo de inferência simbólica (ou outro mecanismo de tratamento simbólico da informação) que irá trabalhar em cooperação com um mecanismo de tratamento conexionista das informações. Este tipo de método é também chamado de hibridização no sentido restrito.

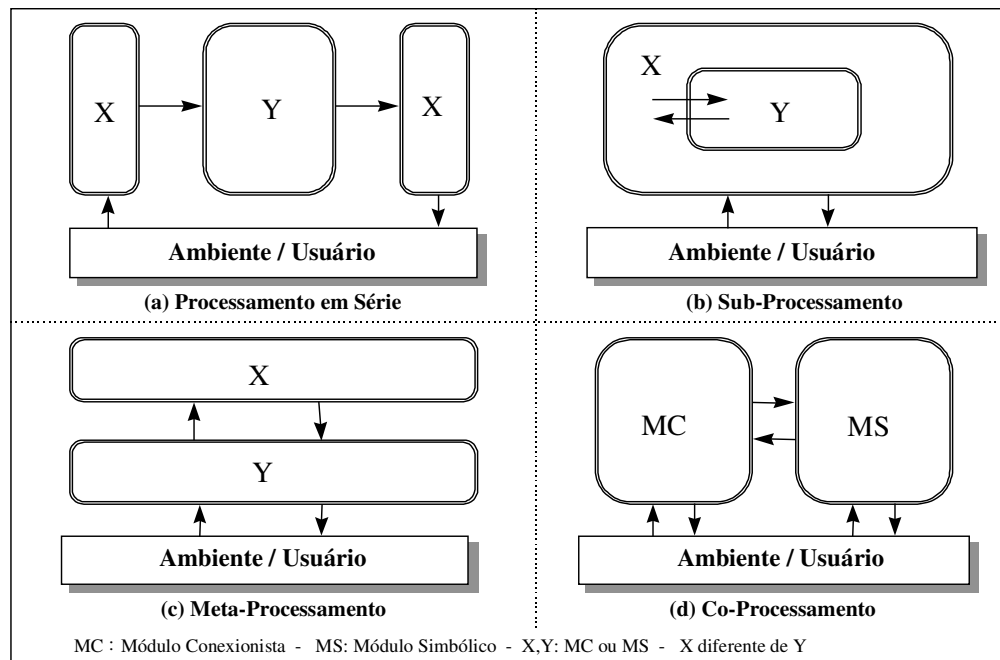


Figura 3.2. Modos de integração Neuro-Simbólica

3.3.1.2. Modos de Integração:

Os modos de integração podem ser classificados nos seguintes grupos (ver Figura 3.2):

1. *Processamento em Série:* O processamento é feito seqüencialmente por um módulo e depois pelo outro (chainprocessing). Por exemplo, um módulo conexionista pode realizar um pré-processamento de dados destinados à um módulo simbólico, ou vice-versa. Este modo de integração é constituído por interações do tipo pré-processamento ou pós-processamento, onde os dois sub-sistemas são ligados por uma simples relação de entrada e saída.
2. *Sub-Processamento:* O processamento principal é realizado por um dos dois módulos, que utiliza o outro módulo apenas como um prestador de serviços para resolver problemas específicos. Por exemplo, um módulo principal simbólico pode disparar a execução de um módulo neural para tratar apenas uma etapa específica do processamento de um problema. Neste tipo de interações, um dos dois módulos é completamente subordinado ao outro, sendo que o processo de controle decide quando e como irá utilizar os resultados obtidos pelo sub-processo por ele executado.
3. *Meta-Processamento:* Um dos dois módulos possui o papel de tratar os problemas em questão, enquanto que o outro módulo assume um papel de nível superior (meta-processamento), podendo ser responsável por intervir em tarefas de controle, observação, gerência ou mesmo melhoria da performance do outro módulo. Um módulo simbólico poderia, por exemplo, controlar e otimizar o funcionamento de uma rede neural.
4. *Co-Processamento:* Os dois módulos são “parceiros iguais” no processo de tratamento de um problema. Neste modo de integração, a palavra-chave é cooperação, onde podemos ter diferentes maneiras de fazer um módulo cooperar com o outro, sem no entanto ser estabelecida uma relação maior de importância ou dependência entre os dois módulos. Uma

maneira de fazer cooperar dois módulos pode ser estabelecida se um módulo se concentra em um aspecto particular do problema em questão, enquanto que o outro módulo se concentra nos outros aspectos distintos deste mesmo problema. Uma outra maneira de cooperar pode ser através de um comportamento concorrente, quando os dois módulos tratam o mesmo problema, mas ao final as duas (ou mais) soluções propostas são avaliadas por um outro módulo a fim de decidir qual é a melhor solução. No modo de co-processamento as trocas de conhecimentos entre os módulos também pode ser importante para que ambos possam cooperar entre si, além também de que normalmente ambos devem também poder interagir com o ambiente externo (usuário/expert).

Este tipo de classificação por modos de integração pressupõe a existência de ao menos dois módulos distintos, um ou mais módulos simbólicos (MS) e um ou mais módulos conexionistas (MC). A figura 3.2 apresenta um esquema a respeito da caracterização dos SHNS em relação ao seu modo de integração.

3.3.1.3. Transferência de Conhecimentos:

As transferências de conhecimentos entre os módulos simbólico e conexionista (ver item sobre 3.3.1.1) podem ser classificadas segundo o sentido (de qual módulo, para qual módulo) das trocas de conhecimentos realizadas:

1. *Do simbólico para o conexionista ($S \Rightarrow C$):* Os conhecimentos são transferidos do módulo simbólico e integrados no módulo conexionista [Boz 95, 97], de modo a permitir a aquisição de conhecimentos por parte do módulo conexionista sem ter que obrigatoriamente recorrer ao aprendizado de uma base de exemplos. Na nossa opinião, os métodos de compilação de regras diretamente sobre a forma de pesos de uma rede neural são os mais interessantes. Existem outras formas de se trocar conhecimentos entre o módulo simbólico e o conexionista, uma destas formas alternativas é a geração automática de exemplos à partir de uma base de regras. Este tipo de procedimento permite que se represente os conhecimentos simbólicos de alto nível através de uma coleção de exemplos que estão de acordo com os conhecimentos representados no sistema simbólico. Um dos inconvenientes desta última técnica é a incerteza de que o módulo conexionista vá conseguir adquirir integralmente todos os conhecimentos que estão sendo repassados através dos exemplos, pois como a rede depende do processo de aprendizado, não há uma garantia que este processo alcance um sucesso 100% de acordo com os exemplos.
2. *Do conexionista para o simbólico ($S \Leftarrow C$):* Os conhecimentos adquiridos através de um processo de aprendizado de uma rede neural, devem ser explicitados sob a forma de conhecimentos simbólicos relativos ao problema que está sendo tratado. A extração de regras à partir de uma rede neural [Andrews 95, 96] permitirá que se utilize estas regras (os conhecimentos adquiridos pelo módulo conexionista) junto ao módulo simbólico. As regras obtidas à partir de uma rede podem ser classificadas de acordo com o tipo de representação simbólica empregado: regras de produção, regras nebulosas, regras probabilistas, etc. O processo de extração de regras pode ser executado considerando-se que podemos tentar extrair todo o conhecimento (o que é muito difícil), ou então, podemos tentar extrair uma parte dos conhecimentos adquiridos pela rede (o que é mais usado – extração dos conhecimentos “mais importantes”).
3. *Transferências bilaterais ($S \Leftrightarrow C$):* Os conhecimentos podem ser transferidos nos dois sentidos entre os módulos simbólico e neural. Os sistemas híbridos com transferências de

conhecimentos bilaterais incluem normalmente mecanismos tanto de compilação de regras dentro de redes neurais, como de extração de conhecimentos destas. Neste caso, podemos ter um módulo simbólico e um módulo conexionista que operam realmente em co-processamento, enquanto que nos demais tipos de transferências, usualmente um dos dois módulos será o principal e o outro atuará apenas como auxiliar. Sempre é bom lembrar a importância que tem o tipo de regras e conhecimentos que serão transferidas de um módulo ao outro, pois isto pode ser um fator de limitação (ou melhoria) importante para o sistema. Concluindo, um sistema com um acoplamento forte e do tipo que usa co-processamento deve obrigatoriamente permitir transferências bilaterais de conhecimentos, onde ambos os módulos devem ser capazes de manipular um tipo “compatível” de representação de conhecimentos usados na transferência.

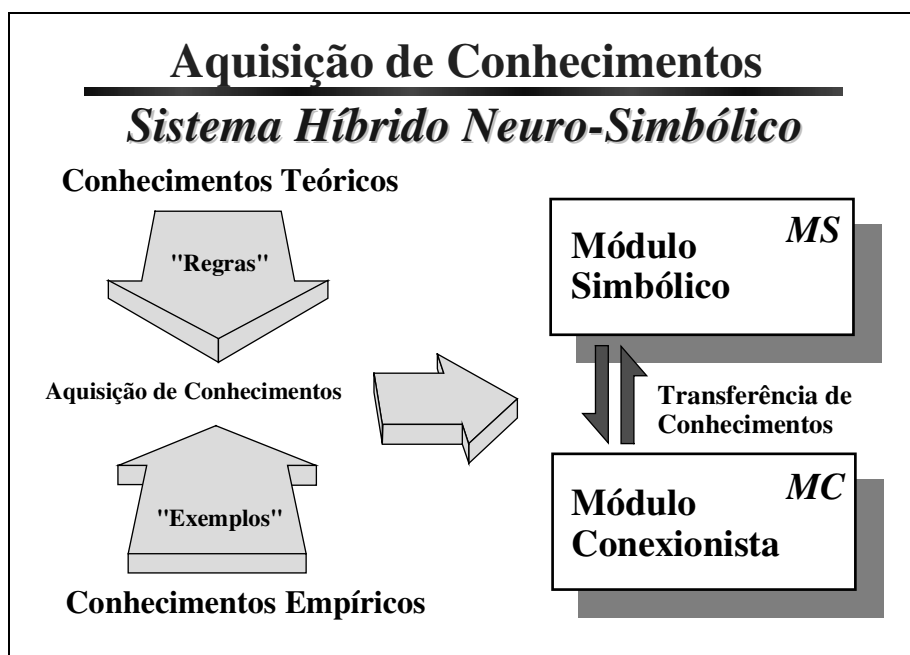


Figura 3.3 – Aquisição e Transferência de Conhecimentos nos SHNS

A capacidade de fazer transferências de conhecimentos entre os módulos de um sistema híbrido é uma das propriedades mais interessantes dos SHNS, pois é através deste tipo de mecanismo que vamos poder fazer interagir e cooperar melhor os módulos simbólicos e conexionistas. Muitas vezes, o processo de inserção de conhecimentos iniciais em uma rede neural vai também contribuir para que o processo de aprendizado seja otimizado (mais rápido e com resultados melhores). A extração de regras à partir das redes neurais vai nos permitir explicitar os conhecimentos adquiridos por esta, e sendo assim poderemos resolver em parte o problema da “caixa preta” que são as redes. As regras extraídas vão permitir também que se compreenda melhor as repostas dadas pelas redes neurais. O módulo simbólico também poderá aproveitar as regras obtidas à partir de uma rede neural, para aumentar a sua base de conhecimentos, onde o processo de extração de regras acaba constituindo-se como mais uma fonte de conhecimentos para o módulo simbólico. Assim esta técnica pode contribuir para minimizar um pouco o problema do gargalo de aquisição de conhecimentos dos sistemas especialistas.

3.4. SHNS: Estudo de Casos

Vamos descrever aqui de maneira bastante sucinta três sistemas híbridos neuro-simbólicos que servem para nos dar uma idéia geral e bastante representativa de como são e como funcionam os SHNS. Os sistemas que escolhemos são: SYNHESYS [Giacometti 92], KBANN [Towell 91] e INSS [Osorio 98]. Existem outros sistemas tão ou talvez mais importantes que estes (e.g. KBCNN [Fu 94], RAPTURE [Mahoney 96], MACIE [Galant 93], CONSYDERR [Sun 95], entre outros), mas nós optamos por discutir apenas sobre estes três, dadas as características que eles possuem, que acreditamos são importantes para que o leitor possa fazer uma análise final do que foi apresentado neste trabalho.

3.4.1. SYNHESYS – Symbolic and Neural Hybrid Expert System Shell

O sistema SYNHESYS foi desenvolvido no Laboratoire LEIBNIZ² em Grenoble-França por Arnaud Giacometti [Giacometti 92]. Este sistema possui um tipo de integração híbrida no sentido estrito, ou seja, ele é realmente composto por dois módulos: um simbólico e outro conexionista. Estes dois módulos operam de maneira cooperativa, permitindo transferências de conhecimentos entre os dois, nos dois sentidos. Portanto este sistema é classificado como um sistema híbrido com transferência de conhecimentos bilaterais e modo de integração por co-processamento. A figura 3.4 apresenta um esquema geral da estrutura de funcionamento do sistema SYNHESYS.

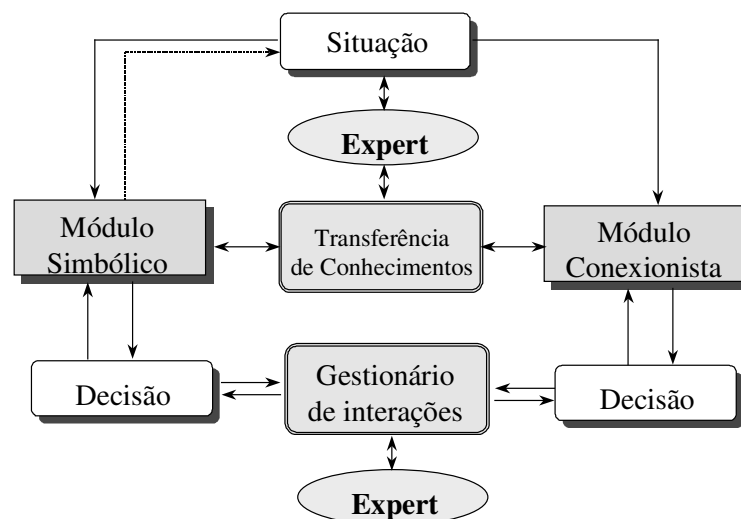


Figura 3.4 – Arquitetura do Sistema SYNHESYS

O módulo simbólico implementa um mecanismo de inferência baseada em regras de produção, com possibilidade de inferência usando “forward chaining” ou “backward chaining”. As regras usadas junto ao módulo simbólico são regras do seguinte tipo:

```
IF <atributo> IN [<valor_inicial_intervalo>,<valor_final_intervalo>] THEN <conclusão>/<ação>
IF <atributo>/<conclusão> AND/OR <atributo>/<conclusão> THEN <conclusão>/<ação>
IF NOT <atributo>/<conclusão> THEN <conclusão>/<ação>
```

O módulo conexionista trabalha com um modelo de rede neural baseado em protótipos de representação localista de conhecimentos (rede do tipo ARN2 desenvolvida por Giacometti e similar em

² Laboratoire LEIBNIZ – Web : <http://www-leibniz.imag.fr/> (Equipe Reseaux d’Automates)

certos aspectos ao modelo ART). Os protótipos criados são hiper-retângulos, o que permite que se codifique as regras descritas acima de maneira bastante simples e direta sob a forma de protótipos da rede. A extração de regras também é feita diretamente à partir dos hiper-retângulos que representam os conhecimentos adquiridos pela rede. Uma outra característica importante deste modelo de rede é o fato que podemos ter um aprendizado contínuo (incremental), pois é possível adicionar novos protótipos à rede sem maiores problemas.

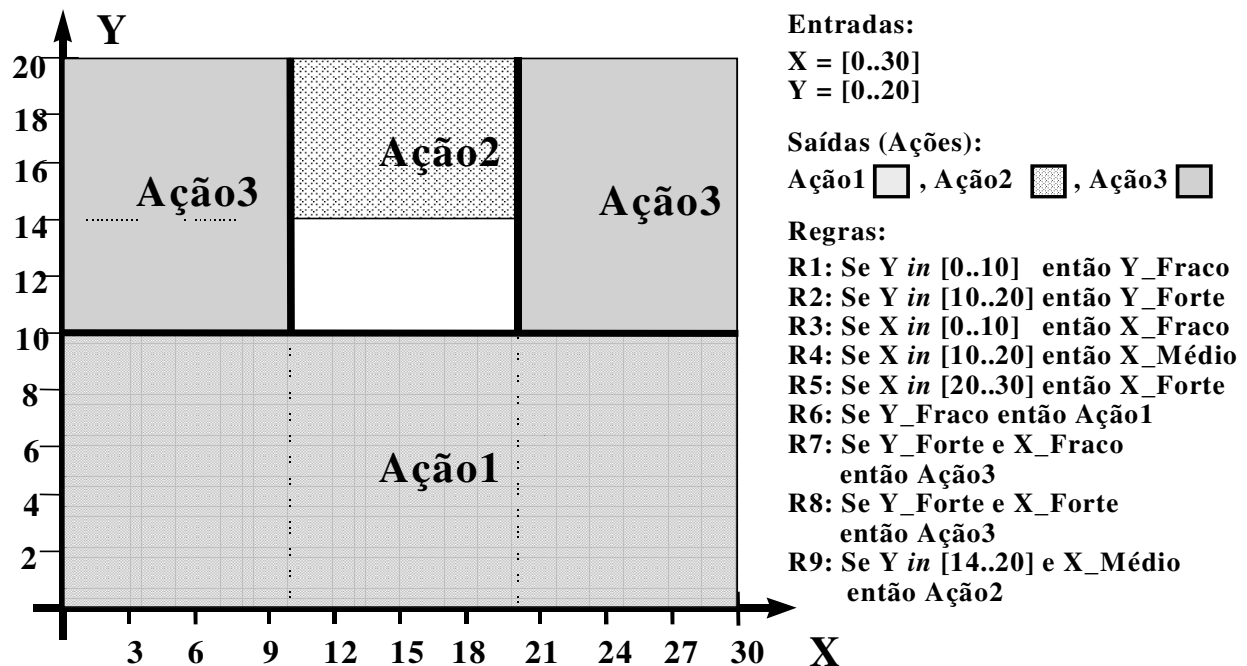


Figura 3.5 – Protótipos do Tipo Hiper-Retângulos e as Regras Correspondentes

O sistema funciona de maneira cooperativa, ou seja, os conhecimentos podem ser adquiridos tanto pelo módulo simbólico (explicitação de regras obtidas junto ao especialista), como também pelo módulo conexionista através do aprendizado à partir de exemplos (criação dos protótipos através de um algoritmo de aprendizado neural). Estes conhecimentos podem ser então compartilhados entre os dois módulos, visto que podemos transferir os conhecimentos adquiridos pelo módulo simbólico para o módulo conexionista, e também no sentido inverso, transferir os conhecimentos adquiridos pelo módulo conexionista para o módulo simbólico.

Ambos os módulos podem compartilhar exatamente os mesmos conhecimentos, ou trabalharem de forma independente. Ao ser realizada uma consulta ao sistema, cada um dos dois módulos vai gerar, independentemente um do outro, uma resposta (decisão). Se ambos os módulos estiverem de acordo em relação a resposta fornecida, esta é repassada para o usuário, e caso não haja um acordo, então é disparado um processo de revisão da base de conhecimentos de ambos os módulos. Inicialmente procura-se identificar a regra/protótipo que está em conflito, e fazendo uso de um processo de interação com o usuário, de transferências de conhecimentos e de identificação de erros, tenta-se desta forma melhorar a base de conhecimentos daquele módulo que apresentou problemas (ou mesmo melhorar ambos os módulos).

Os princípios que regem o funcionamento deste sistema são um exemplo extremamente significativo de como pode ser explorada esta integração neuro-simbólica em um sistema híbrido. O sistema SYNHESYS foi usado com sucesso em diversas aplicações, mas no entanto, apesar de suas excelentes

características e potenciais, foram encontrados alguns problemas e limitações no que se refere ao tratamento de problemas onde se tenha o uso de dados do tipo qualitativo [Orsier 94, 95]. Estes problemas são relativos as métricas para o cálculo de similaridade entre exemplos e protótipos, que funcionam melhor com dados quantitativos do que com dados qualitativos. Apesar disto, acreditamos que o sistema SYNHESYS continua sendo um dos melhores exemplos de integração neuro-simbólica existentes.

3.4.2. KBANN – Knowledge Based Artificial Neural Networks

O sistema KBANN foi desenvolvido no UWM-MLRG³ da Universidade de Wisconsin-Madison por Geoffrey Towell e Jude Shavlik [Towell 91]. Este sistema possui apenas um módulo conexionista, que como diz o seu nome, é uma rede neural baseada em conhecimentos. O sistema KBANN possui uma rede neural capaz de adquirir conhecimentos simbólicos expressos através de regras, realizar o aprendizado neural e depois extrair regras que representam o conhecimento que foi “refinado” pelo aprendizado de exemplos. A proposta original do sistema KBANN não previa outros modos de interação com módulos simbólicos externos à rede, que fossem adicionais à troca de conhecimentos realizada através da inserção e extração de regras à partir destas redes neurais. A figura 3.6 mostra um esquema da forma de funcionamento do sistema KBANN.

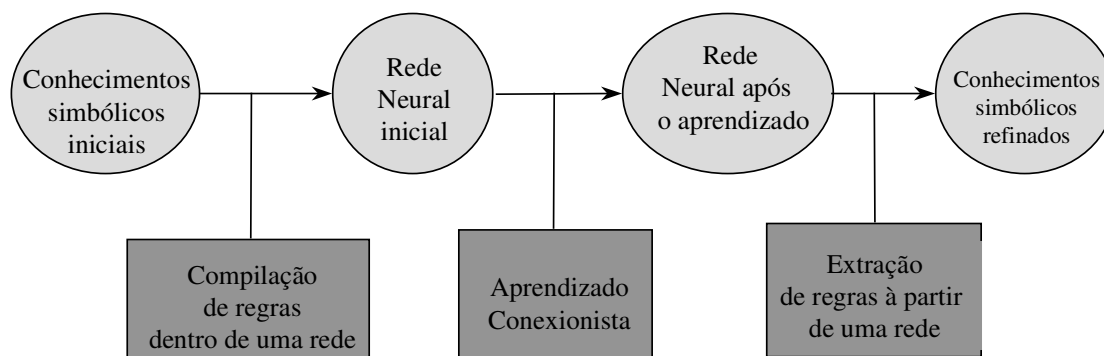


Figura 3.6 - Arquitetura do Sistema KBANN

Sendo assim, as transferências de conhecimentos entre os dois podem ser realizadas nos dois sentidos, apesar de apenas o módulo conexionista ter sido definido. Portanto este sistema pode ser classificado como um sistema híbrido com transferência de conhecimentos bilaterais e modo de integração por co-processamento. Ele também pode ser considerado como um sistema puramente conexionista (híbrido no sentido amplo), pois prevê que todo o tratamento de informações é centralizado e realizado pelo módulo conexionista. Os conhecimentos simbólicos, obtidos usualmente à partir da explicitação de regras realizada por um especialista, são integrados na rede neural, sem termos a necessidade de utilizar um módulo simbólico de processamento. A extração de regras tem como objetivo a explicitação dos conhecimentos adquiridos pela rede, permitindo que se compreenda como a rede vai gerar suas respostas. Estas regras até poderiam ser usadas por um módulo simbólico, mas no entanto esta não é a principal função das regras simbólicas neste sistema.

³ UWM-MLRG – University of Wisconsin-Madison / Machine Learning Research Group.
Web: <http://www.cs.wisc.edu/~shavlik/uwml.html>

O sistema KBANN utiliza redes neurais do tipo MLP (*Multi-Layer Perceptron*) com aprendizado através do uso do algoritmo do *Back-Propagation* com mínimas modificações. Este modelo de rede neural é muito usado atualmente, mas possui algumas limitações bem conhecidas em relação à velocidade do aprendizado e a otimização dos pesos da rede (discutido no capítulo anterior, seção 2.2). Os aspectos mais interessantes do modelo proposto por Towell dizem respeito às técnicas de inserção e extração de regras, usando como elemento de base redes do tipo MLP.

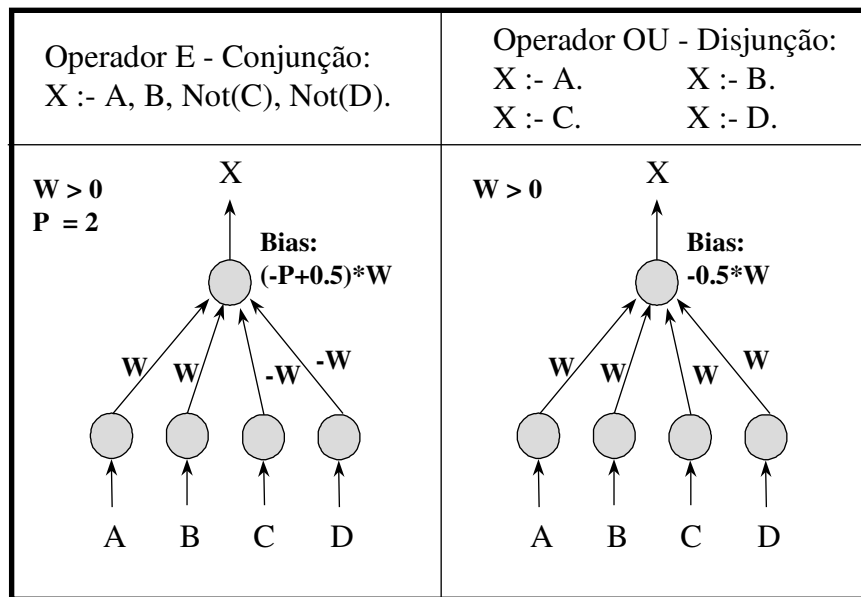


Figura 3.7 – Compilação de Regras: Inserção de Conhecimentos Simbólicos em Redes MLP

A compilação de regras [Towell 91, 94] permite que se obtenha uma rede neural que vai dar respostas exatamente iguais aquelas que nós poderíamos obter caso fosse utilizado um mecanismo de inferência simbólica com este mesmo conjunto de regras. Desta maneira vamos poder criar a estrutura básica e especificar os pesos iniciais da rede de forma automática, dotando assim a rede de conhecimentos iniciais, sem no entanto termos que recorrer aos procedimentos tradicionais de aprendizado de exemplos por adaptação dos pesos da rede. A figura 3.7 mostra um exemplo de compilação de regras em uma rede neural.

As regras inseridas nas redes do tipo KBANN são basicamente regras de produção que manipulam valores binários relativos aos atributos de entrada (antecedentes) e gerando também conclusões (conseqüentes) binárias do tipo verdadeiro/falso. As regras usadas são portanto do seguinte tipo:

IF <antecedente> *THEN* <conseqüente>
IF <antecedente> *AND* / *OR* <antecedente> *THEN* <conseqüente>
IF NOT <antecedente> *THEN* <conseqüente>

Este conhecimento inicial que foi introduzido na rede passa por uma fase de refinamento, que é realizada através da utilização do algoritmo de aprendizado neural. A rede KBANN passa por um processo de adaptação dos pesos da rede através do uso de uma base de exemplos de aprendizado, da mesma forma como é feito com outros modelos de redes neurais, mas a principal diferença é que a rede utilizada foi inicializada com a inserção de conhecimentos *a priori* disponíveis sobre o problema. Esta inserção de conhecimentos permite uma otimização do processo de aprendizado, bem como o aproveitamento dos diferentes conhecimentos disponíveis sobre o problema, sejam eles representados sob

a forma de regras simbólicas (conhecimentos teóricos) ou sob a forma de exemplos (conhecimentos empíricos).

Uma vez que a rede passou por esta etapa de adaptação, podemos então realizar a extração de regras simbólicas à partir de uma análise dos pesos da rede. O processo de extração de regras vai nos permitir a obtenção de regras do mesmo tipo daquelas que foram inseridas na rede, mas que usualmente devem refletir as “melhorias” feitas durante o processo de aprendizado de uma base de exemplos. O processo de extração de regras necessita de um grande processamento de informações, tendo um alto custo computacional, mas por outro lado, trazendo grandes vantagens devido a tradução do conhecimento neural sob uma forma de representação bem mais compreensível ao ser humano. Towell apresenta basicamente dois métodos de extração de regras à partir das redes KBANN, o SubSet e o NofM [Towell 91, 93], onde representamos o primeiro deles de maneira esquemática na figura 3.8.

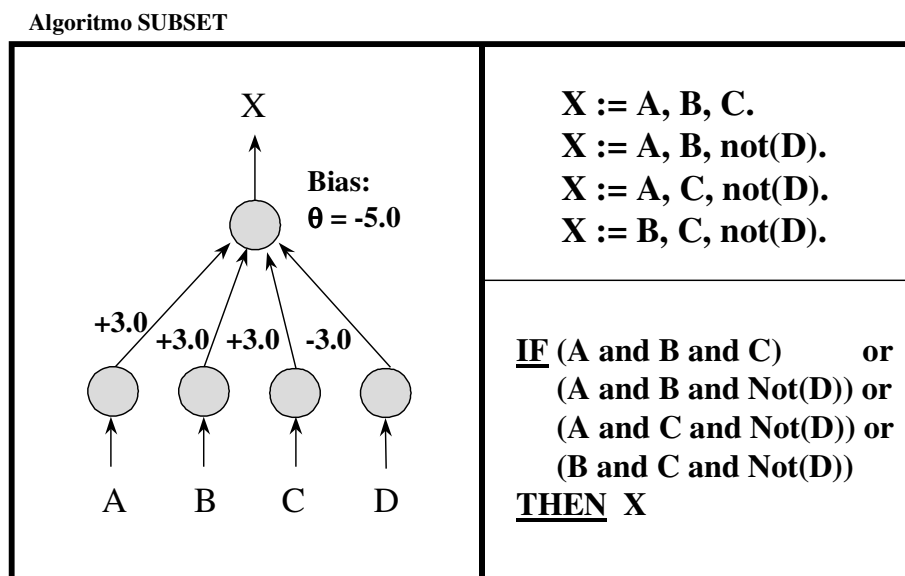


Figura 3.8 – Extração de Regras à partir de uma Rede KBANN

As redes KBANN possuem uma característica interessante que é a possibilidade de fechar o ciclo de refinamento de regras, com a re-inserção das regras extraídas dentro da rede neural, o que permitiria um processamento evolutivo dos conhecimentos sobre o problema. Entretanto, este tipo de procedimento não foi estudado em profundidade e os testes e estudos sobre este tema que foram realizados são muito superficiais. Por outro lado, as redes KBANN possuem algumas importantes limitações relativas ao tipo de algoritmos de aprendizado, compilação e extração de regras que estas utilizam [Osorio 98, 98a, 99]. Outro ponto importante a ser considerado, é referente ao tipo de regras manipuladas pelas redes KBANN, que não são adequadas ao tratamento de dados quantitativos, ficando muito restritas aos dados qualitativos. E finalmente, é bom ressaltar que as redes KBANN *não* são redes do tipo incremental.

3.4.3. INSS – Incremental Neuro-Symbolic System

O sistema INSS foi desenvolvido no Laboratoire LEIBNIZ⁴ em Grenoble-França por Fernando Osório e Bernad Amy [Osorio 98, 99a]. Este sistema possui um tipo de integração híbrida no sentido estrito, ou seja, ele pressupõe a utilização de dois módulos: um simbólico e outro conexionista. Estes dois

⁴ Laboratoire LEIBNIZ – Web : <http://www-leibniz.imag.fr/> (Equipe Reseaux d'Automates)

módulos operam de maneira cooperativa, permitindo transferências de conhecimentos entre os dois, nos dois sentidos. Portanto este sistema é classificado como um sistema híbrido com transferência de conhecimentos bilaterais e modo de integração por co-processamento. A figura 3.4 apresenta um esquema geral da estrutura de funcionamento do sistema INSS.

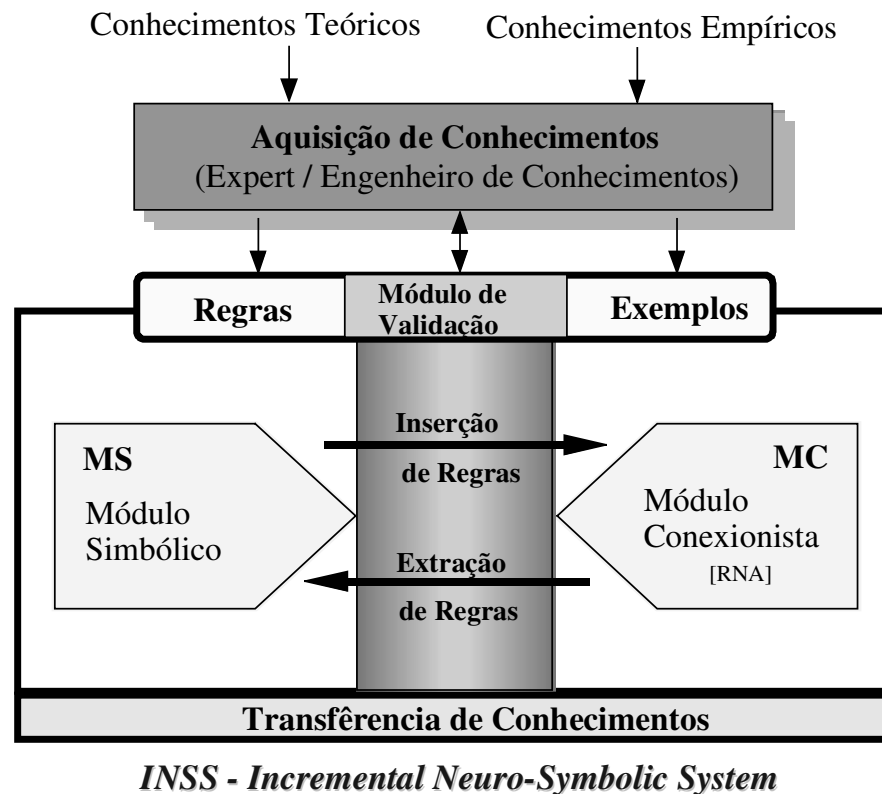


Figura 3.9 – Esquema de Integração Neuro-Simbólica do Sistema INSS

O desenvolvimento do sistema INSS foi bastante influenciado pelos dois outros sistemas que descrevemos anteriormente, o sistema SYNHESYS e o sistema KBANN. O modo de operar é bastante similar ao sistema SYNHESYS, ou seja, temos dois módulos que podem trocar conhecimentos entre si, através da inserção de regras simbólicas no módulo conexionista e através da extração de regras deste mesmo módulo. Além disto, foi implementada uma aquisição incremental de conhecimentos que permite adicionar novos conhecimentos na rede (aprendizado contínuo), alterar a estrutura da rede neural adicionando novos neurônios (estrutura incremental), extrair apenas os novos conhecimentos adquiridos (extração incremental) e validar os resultados e conhecimentos. O tipo de regras usadas junto ao sistema INSS também é mais próximo às regras simbólicas usadas pelo sistema SYNHESYS, onde podemos inserir regras nas redes que contêm operadores como:

Greater_Than (Entrada, Valor_Constante);
Greater_Than (Entrada1, Entrada2)
Less_Than (Entrada, Valor_Constante)
Less_Than (Entrada1, Entrada2)
In_Range (Entrada, Valor_Constante1, Valor_Constante2)
Equal (Entrada, Valor_Constante).

O módulo conexionista possui muitas semelhanças ao modelo adotado pelas redes KBANN, pois adotamos um modelo de rede do tipo MLP (*Multi-Layer Perceptron*), mas no lugar do algoritmo de aprendizado do tipo *Back-Propagation* das redes KBANN, foi utilizado o algoritmo *Cascade-Correlation*. Este algoritmo confere ao sistema INSS alguma de suas principais características:

- O algoritmo de aprendizado tem um ótimo desempenho (velocidade, capacidade de aprendizado);
- Este algoritmo permite a inserção de novos neurônios na rede durante o aprendizado, o que permite um aprendizado incremental (ao nível dos dados e da estrutura da rede);
- Permite a inserção de conhecimentos a priori na rede que permanecem inalterados, onde as novas unidades adicionadas na rede é que terão por função melhorar ou corrigir o comportamento desta;
- Como os conhecimentos previamente adquiridos não são modificados, e apenas inserimos novos conhecimentos complementares a estes, ao realizar a extração de regras, podemos nos concentrar apenas nos novos conhecimentos adquiridos. Não é necessário re-extrair todas as regras que já haviam sido inseridas anteriormente na rede.

O módulo simbólico do sistema INSS utiliza o motor de inferência CLIPS como base para o processamento simbólico. Este sistema vem sendo substituído pelo conceito de que a própria rede neural pode implementar um motor de inferência uma vez que as regras simbólicas foram inseridas nela. O mecanismo de inferência permite apenas o “*forward chaining*”, mas existe a possibilidade de expandir este tipo de inferência, permitindo também o “*backward chaining*”.

O sistema INSS está organizado em módulos que realizam as diferentes tarefas necessárias para o bom funcionamento do sistema. Os módulos que compõem este sistema são: *CLIPS* (Módulo Simbólico), *NeuSim* (Módulo Neural – Simulador / Aprendizado), *NeuComp* (Compilador de regras simbólicas em redes), *Extract* (Extração de regras à partir das redes neurais) e *Valid* (Módulo de verificação e validação dos conhecimentos do sistemas, representados pelos exemplos e regras). A figura 3.10 mostra um diagrama esquemático dos componentes do sistema.

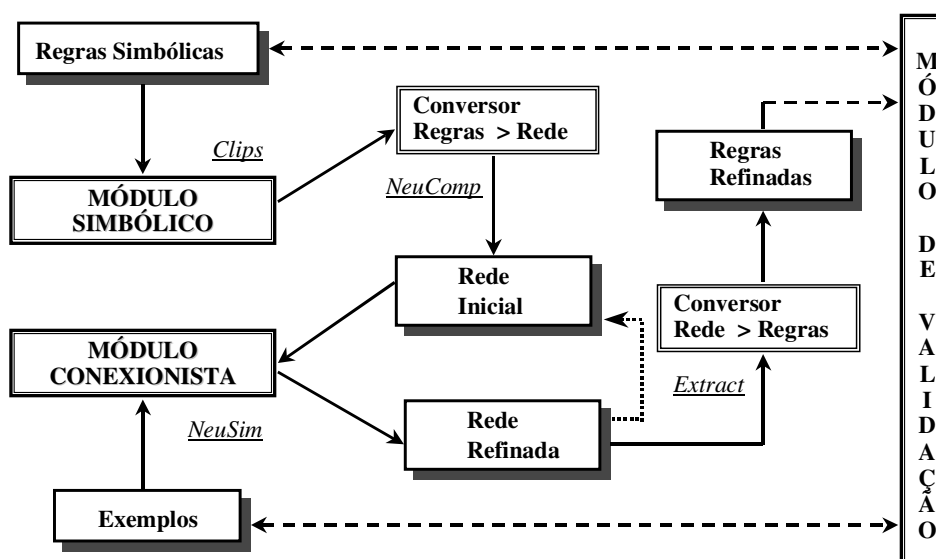


Figura 3.10 – Módulos do Sistema INSS

O módulo compilador de regras NeuComp utiliza o mesmo algoritmo de inserção de regras usado nas redes KBANN, mas porém este possui uma extensão própria, que foi desenvolvida especificamente para este sistema a fim de lhe permitir o tratamento das regras tipo Greater_Than, Less_Than, In_Range e Equal. O módulo de extração de regras Extract é baseado também nos algoritmos usados nas redes KBANN, o SubSet e o NofM. Esta é uma das principais limitações do sistema INSS, pois não devido ao uso destes algoritmo não podemos extrair regras com o mesmo poder de representação de conhecimentos das regras que podemos inserir no sistema. Estudos vem sendo desenvolvidos no sentido de resolver esta deficiência.

O sistema INSS foi aplicado em diferentes tipos de problemas, com resultados muito satisfatórios [Osorio 98]. Os problemas tratados foram: *Monk's Problem* (Benchmark de algoritmos de aprendizado de máquinas), Diagnóstico médico (Identificação da causa de comas tóxicos – Projeto Europeu Esprit MIX), Robótica Autônoma (Comportamento Sensorial-Motor – Evitar obstáculos, acompanhar uma parede) e Modelagem Cognitiva usando o problema da balança (*Balance Scale Problem*).

3.5. Considerações Finais

Este trabalho teve por objetivo mostrar a importância da utilização de sistemas híbridos quando se deseja implementar um sistema inteligente. Vamos resumir aqui os pontos principais que gostaríamos que fossem salientados em relação a este tipo de sistemas.

Para que um sistema possa ser dotado de propriedades que permitam que ele possa ser classificado como “inteligente”, existem alguns itens importantes que devem ser levados em consideração:

- Aquisição de conhecimentos: O processo de aquisição de conhecimentos é vital para um sistema inteligente. A aquisição de conhecimentos deve permitir que se use todos os conhecimentos disponíveis, sejam eles conhecimentos básicos e empíricos (casos práticos e exemplos), sejam eles conhecimentos teóricos já tratados e estruturados (regras e outros conhecimentos simbólicos), ou mesmo conhecimentos que possam vir de outros agentes.
- Representação de conhecimentos: O poder de representação e manipulação de conhecimentos irá definir as principais limitações de um sistema. Uma das mais poderosas ferramentas de representação de conhecimentos é a linguagem natural humana, associada a fala e a escrita (com todas as suas extensões, tal como a simbologia matemática). Assim como a comunicação humana é rica e pluralista, os conhecimentos representados em um sistema inteligente também deverão ser. Conhecimentos devem poder ser aproximados, inexatos e mesmo contraditórios.
- Múltiplos mecanismos de tratamento dos conhecimentos (hybrid reasoning) e modularidade: A cooperação entre sistemas onde cada um pode complementar o que o outro faz, ou onde as tarefas podem ser divididas e distribuídas segundo as especificidades de cada módulo, são a chave para o desenvolvimento de sistemas capazes de tratar problemas muito complexos.
- Transferência de conhecimentos e interação entre módulos: Para que sistemas possam interagir e cooperar é importante que tenhamos uma poderosa forma de transferência de conhecimentos. O ser humano tem evoluído suas capacidades intelectuais e produtivas de maneira espantosa, onde indiscutivelmente um elemento de suma importância neste processo é a sua capacidade de trocar conhecimentos e interagir com os demais seres humanos.
- Validação e evolução dos conhecimentos: Os conhecimentos adquiridos devem ser constantemente validados, pois é justamente do conflito cognitivo oriundo desta validação

(conhecimentos anteriores *versus* conhecimentos novos) que podem surgir novas melhorias dos conhecimentos. Quando é constatado que uma certa teoria entra em conflito com a prática, é justamente baseado neste conflito que um pesquisador irá buscar uma nova reformulação desta. As teorias devem estar sendo constantemente questionadas e revisadas perante os fatos e experiências do nosso dia-à-dia. Os conhecimentos estão sempre evoluindo.

Concluindo, os sistemas híbridos são uma tendência atual nos estudos referentes a Inteligência Artificial, sejam eles denominados simplesmente Sistemas Híbridos Inteligentes, ou de Sistemas Multi-Agentes, ou ainda de Inteligência Artificial Distribuída. Acreditamos que a base de tais sistemas, que deve estar sempre no centro das pesquisas realizadas nesta área, é relativa a alguns pontos que abordamos de forma insistente neste trabalho: a capacidade de representação de conhecimentos (preferencialmente múltiplos), a realização de trocas de conhecimentos entre componentes do sistema (onde a representação volta a ser um item central), e a evolução dos conhecimentos (aquisição e refinamento progressivos).

HYBRID MACHINE LEARNING TOOLS



BIBLIOGRAFIA

[Alche 93] d'ALCHÉ-BUC, Florence. *Modèles Neuronaux et Algorithmes Constructifs pour l'Apprentissage de Règles de Décision*. Thèse de Doctorat en Sciences. Laboratoire LEP, Université de Paris-Sud XI - Centre d'Orsay, France. Décembre 1993.

[Andrews 95] ANDREWS, R.; DIEDERICH, J.; TICKLE, A.B. *A Survey And Critique of Techniques For Extracting Rules From Trained ANN*. Technical Report - Neurocomputing Research Centre, Queensland University of Technology - Brisbane, Australia. January 1995 (To appear in : Knowledge-Based Systems).
Web: <http://www.fit.qut.edu.au/NRC/>
Ftp: <ftp://ftp.fit.qut.edu.au/pub/NRC/tr/ps/QUTNRC-95-01-02.ps.Z>

[Andrews 96] ANDREWS, Robert & DIEDERICH, Joachim. (Eds.) *Rules and Networks - Proceedings of the Rule Extraction From Trained Artificial Neural Networks Workshop*. SSAISB-AISB'96, University of Sussex, Brighton, UK. 2nd April. Published by QUT - Queensland Univ. of Technology, Australia. 1996.

[Balakrishnan 95] BALAKRISHNAN, Karthik & HONAVAR, Vasant. *Evolutionary Design of Neural Architectures - A Preliminary Taxonomy and Guide to Literature*. Technical Report: ISU CS-TR-95-01,

AI group - Dept. Computer Science, Iowa State University. 1995. Web: <http://www.cs.iastate.edu/~honavar/publist.html>

[Boz 95] BOZ, Olcay. *Knowledge Integration and Rule Extraction in Neural Networks*. Technical Report, EECS, Lehigh University. October 1995.
Web: <http://www.lehigh.edu/~ob00/integrated.html>

[Boz 97] BOZ, Olcay. *Converting a Trained Neural Net to a Decision Tree - DECEXT (Decision Tree Extractor)*. Technical Report - University of Lehigh, U.S.A.
Web: <http://www.lehigh.edu/~ob00/integrated.html>

[Carpenter 92] CARPENTER, G. et al. *Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps*. IEEE Transactions on Neural Networks 3(5), pp.698-713. September 1992.
Web: <http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/fuzzy/systems/artmap/>

[Cechin 98] CECHIN, Adelmo. *The Extraction of Fuzzy Rules from Neural Networks*. Shaker Verlag, Berlin. Tese de doutorado – Universidade de Tübingen. 1998.

[Cox 92] COX, Earl. *Integrating Fuzzy Logic into Neural Nets*. AI Expert, June 1992.

[Craven 96] CRAVEN, M. & SHAVLIK, J. *Extracting Tree-Structured Representations of Trained Networks*. Advances in Neural Information Processing Systems, MIT Press, Denver. pp.24-30. 1996.

[Craven 96a] CRAVEN, Mark W. *Extracting Comprehensible Models from Trained Neural Networks*. Ph.D. Thesis - Dept. of Computer Science, Univ. of Wisconsin-Madison, U.S.A. 1996. Web: <http://www.cs.wisc.edu/~shavlik/mlrg/publications.html>

[DeJong 93b] DE JONG, K.; SPEARS, W.; GORDON, D. *Usign Genetic Algorithms for Concept Learning*. Machine Learning, 13, pp.161-187. Kluwer Academic Publishers, Boston, MA - U.S.A. 1993.

[Eick 96] EICK, Christoph F. ; KIM, Yeong-Joon ; SECOMANDI, Nicola ; TOTO, Ema. *DELVAUX - An Environment that Learns Bayesian Rule-Sets with Genetic Algorithms*. In : The Third World Congress on Expert Systems. Seoul, Korea, February, 1996. Web: <http://www.cs.uh.edu/~yjkim/> or <http://www.cs.uh.edu/~ceick/ceick.html>

[Faq 99b] FAQ. *Faq Fuzzy Logic & Fuzzy Systems - Comp.ai.fuzzy..* 1999.
Web: <http://www.cis.ohio-state.edu/hypertext/faq/usenet/fuzzy-logic/part1/faq.html>

[Fu 94] FU. LiMin. *Neural Networks in Computer Intelligence*. McGraw-Hill Publishing Inc., 1994.

[Gallant 88] GALLANT, S. I. *Connectionist Expert Systems*. Communications of the ACM, 31(2), pp.152-169. February 1988.

[Gallant 93] GALLANT, S. I. *Neural Network Learning and Expert Systems*. MIT Press, Cambridge, MA. 1993.

[Giarratano 93] GIARRATANO, Joseph C. *CLIPS User's Guide - Version 6.0*. Lyndon B. Johnson Space Center, Software Technology Branch, NASA - U.S.A. 1993. Web: <http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/expert/systems/clips/> or <http://home.haley.com/clips.html>

[Giacometti 92] GIACOMETTI, Arnaud. *Modèles Hybrides de l'Expertise*. Thèse de Doctorat en Informatique et Réseaux, Lab. LIFIA - IMAG, Grenoble / ENST Paris - France, Novembre 1992. Web: <http://www-leibniz.imag.fr/RESEAUX/public.html> .

Ftp: <ftp://ftp.imag.fr/pub/LEIBNIZ/ESEAUX-D-AUTOMATES/giacometti.these.ps.tar.gz>

[Gonzalez 96] GINZALEZ, J.C.; VELASCO, J.R.; IGLESIAS, C.A. *Prototype of a Fuzzy-Neural Hybrid Model*. Evaluation report. Rapport Technique MIX/WP2/UPM/3.0, UPM, MIX Project deliverable D7. 1996.

[Greene 93] GREENE, David P. & SMITH, Stephen, F. *Competition-Based Induction of Decision Models from Examples*. Machine Learning, 13, pp.229-258. Kluwer Academic Publishers, Boston, MA - U.S.A. 1993.

[Halgamuge 94] HALGAMUGE, S.K. *FuNeGen - Fuzzy Neural System*. Publicado na Internet. Web: <http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/fuzzy/systems/funegen/>

[Hayashi 94] HAYASHI, Y. & BUCKLEY, J. J. *Approximations Between Fuzzy Expert Systems and Neural Networks*. International Journal of Approximate Reasoning, Vol.10, pp.63-73. 1994

[Hilario 93] HILARIO, Melanie. *MIX: Modular Integration of Connectionist and Symbolic Processing in Knowledge-Based Systems*. Proposal for Basic Research Project - Esprit BRA, EEC, n° 09119.

[Hilario 94] HILARIO, M.; PELLEGRINI, C.; ALEXANDRE, F. *Modular Integration of Connectionist and Symbolic Processing*. In : Knowledge-Based Systems - Proceedings of the ISIKNH'94: International Symposium on Integrating Knowledge and Neural Heuristics, pp. 123-132. Pensacola, Florida: May 1994. Ftp: <ftp://cui.unige.ch/AI/mix/isiknh-94.ps>

[Hilario 96] HILARIO, M. *An Overview of Strategies for Neurosymbolic Integration*. In: Connectionist-Symbolic Integration: From Unified to Hybrid Approaches. Ron Sun (Ed.) - Chapter 2. Kluwer Academic Publishers, 1996. Ftp: <ftp://cui.unige.ch/AI/>

[IIT 96] ITI - Institute for Information Technology. *Fuzzy CLIPS*. Publié dans le WEB-Internet. IIT - NRC, Canada. Web: <http://ai.iit.nrc.ca/fuzzy/fuzzy.html>
Ftp: <ftp://ftp.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/expert/systems/clips/0.html>

[Jang 93] JANG, Jyh-Shing. *ANFIS: Adaptive-Network-Based Fuzzy Inference Systems*. IEEE Transactions on Systems, Man and Cybernetics. May 1993.
Web: <http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/fuzzy/systems/anfis/>

[Jang 96] JANG, J.-S. R.; SUN, C.-T.; MIZUTANI, E. *Neuro-Fuzzy and Soft Computing*. Prentice Hall, 1997.

[Kasabov 90] KASABOV, Nikola K. *Hybrid Connectionist Rule-Based Systems*. In: Artificial Intelligence IV: Methodology, Systems, Applications. Proceedings of AIMS'90 - Albena, Bulgaria. P. Jorrand and V. Sgurev (Eds.). Amsterdam, APIA - Elsevier Science Publishers. September 1990.

[Kasabov 95] KASABOV, N. *Learning Fuzzy Rules and Approximate Reasoning in Fuzzy Neural Networks and Hybrid Systems*. Fuzzy Sets and Systems. Special Issue, pp.1-19. 1995. Web: <http://divcom.otago.ac.nz:800/COM/INFOSCI/KEL/>

[Kasabov 96] KASABOV, N.K. *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engeneering*. Cambridge, Massachussets, MIT Press. 1996.

Web: <http://divcom.otago.ac.nz:800/COM/INFOSCI/KEL/>

[Kilgour 97] KILGOUR, Richard. *FuzzyCOPE 1, FuzzyCOPE2 and FuNN*. Publié dans le WEB-Internet.

Web: <http://divcom.otago.ac.nz:800/COM/INFOSCI/KEL/software.htm>

[Krzysztof 97] KRZYSZTOF, J. C. & PEDRYCZ, W. *Neuro-Fuzzy Systems*. In: Handbook of Neural Computation (section D1). E. Fiesler and R. Beale (Eds.) Institute of Physics and Oxford University Press. New York, NY - U.S.A., 1997. Web: <http://www.idiap.ch/publications/fiesler-96.1.bib.abs.html> or http://www.oup-usa.org/acadref/nc_accs.html

[Lee 95] LEE, S.-J. & JONE, M.-T. *A Symbolic Logic Approach of Deriving Initial Neural Network Configurations for Supervised Classification*. Computers and Artificial Intelligence, vol.14(4), pp.317-337. 1995.

[Machado 92] MACHADO, Ricardo J. & ROCHA, A.F. *A hybrid Architecture for Fuzzy Connectionist Expert System*. In: Hybrid Architectures for Intelligent Systems. A. Kandel and G. Langholz (Eds.). CRC Press, Boca Raton - Florida, 1992.

[Mahoney 93] MAHONEY, J. & Mooney, R. *Combining Connectionist and Symbolic Learning to Refine Certainty-Factor Rule-Bases*. Connection Science, 5(1993), pp.339-364. 1993.

Web: <http://net.cs.utexas.edu/users/ml/> . Ftp: <ftp://ftp.cs.utexas.edu/pub/mooney/papers>

[Mahoney 96] MAHONEY, J. Jeffrey. *Combining Symbolic and Connectionist Learning Methods to Refine Certainty-Factor Rule Bases*. Ph.D. Thesis, Dept. of Computer Science, Univ. of Texas at Austin, May 1996. Web: <http://net.cs.utexas.edu/users/ml/> . Ftp: [ftp://ftp.cs.utexas.edu/pub/mooney/papers\(rapture-dissertation-96.ps.Z\)](ftp://ftp.cs.utexas.edu/pub/mooney/papers/rapture-dissertation-96.ps.Z)

[Malek 95] MALEK, Maria & LABBI, A. *Integration of Case-Based Reasoning and Neural Approaches for Classification*. Rapport Technique - RT 131, LIFIA, IMAG. Grenoble, France. 1995. Web: <http://www-leibniz.imag.fr/RESEAUX/public.html>

[Malek 96] MALEK, Maria, AMY B. *A Preprocessing Model for Integrating Case-Based Reasoning and Prototype-Based Neural Network*. Connectionist Symbolic Integration, Lawrence, Erlbaum Associates. 1996. Web: <http://www-leibniz.imag.fr/RESEAUX/public.html>

[Malek 96a] MALEK, Maria. *Un modèle hybride de mémoire pour le raisonnement à partir de cas*. Thèse de Doctorat en Informatique, Lab. LEIBNIZ - Université Joseph Fourier, IMAG, Grenoble - France, Octobre 1996. Web: <http://www-leibniz.imag.fr/RESEAUX/public.html> . Ftp: <ftp://ftp.imag.fr/pub/LEIBNIZ/RESEAUX-D-AUTOMATES/malek.these.ps.gz>

[Medsker 95] MEDSKER, L.R. *Hybrid Intelligent Systems*. Kluwer Academic Publishers, Boston. 1995.

[Mitchell 97] MITCHELL, Tom M. *Machine Learning*. McGraw-Hill Publishing Company, McGraw-Hill Series in Computer Science (Artificial Intelligence). 1997.

[Nauck 95] NAUCK, Detlef. *Beyond Neuro-Fuzzy: Perspectives and Directions*. In: Proceedings of the Third European Congress on Intelligent Techniques and Soft Computing - EUFIT'95, Aachen, pp.1159-1164. August 1995. Web: <http://www.cs.tu-bs.de/~nauk>

[Nauck 97] NAUCK, D.; NÜRNBERGER, A.; KRUSE, R. *NEFCON, NEFCLASS and NEFPROX*. Publié dans le WEB-Internet. Web: <http://fuzzy.cs.uni-magdeburg.de/nmfuz.html> or <http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/fuzzy/systems/nefcon>

[Nolfi 94] NOLFI, Stefano; PARISI, Domenico; ELMAN, Jeffrey L. *Learning and Evolution in Neural Networks*. Adaptive Behavior, 3(1), pp.5-28. 1994. Web: <http://crl.ucsd.edu/~elman/>

[Orsier 94] ORSIER, B.; Amy, B.; Rialle, V.; Giacometti, A. *A Study of the Hybrid System SYNHESYS*. In: Proceedings of the ECAI-94 Workshop - Combining Symbolic and Connectionist Processing, August 1994 - Amsterdam. 1994.

Web: <http://cuiwww.unige.ch/AI-group/staff/orsier.html>

[Orsier 95] ORSIER, Bruno. *Etude et Application de Systèmes Hybrides NeuroSymboliques*. Thèse en Informatique, Laboratoire LIFIA-IMAG, UJF - Grenoble, 1995.

Web: <http://www-leibniz.imag.fr/RESEAUX/public.html>

Ftp: <ftp://ftp.imag.fr/pub/LEIBNIZ/RESEAUX-D-AUTOMATES/orsier.these.ps.gz>

[Osório 95] OSORIO, F.S. e AMY, Bernard. *INSS: Um Sistema Híbrido Simbólico-Connexionista com Aprendizado à partir de Regras e de Exemplos*. Panel'95 - XXI Conferencia Latino-Americana de Informática, Canela, Brasil. August 1995.

Web: <http://www-leibniz.imag.fr/RESEAUX/osorio/articles/diret.html>

[Osorio 98] OSORIO, F. S. *INSS : Un Système Hybride Neuro-Symbolique pour l'Apprentissage Automatique Constructif*. Tese de Doutorado (Ph.D. Thesis). Laboratoire LEIBNIZ – IMAG / INPG, Grenoble – França, 1998. Web: <http://www-leibniz.imag.fr/RESEAUX/public.html> ou <http://www-leibniz.imag.fr/RESEAUX/osorio.html>.

Ftp: <ftp://ftp.imag.fr/pub/LEIBNIZ/RESEAUX-D-AUTOMATES/osorio.these.ps.gz>

[Osorio 98a] OSORIO, F. S., AMY, Bernard ; DECLOEDT, Loïc. Rule-Out: Um novo método de extração incremental de regras à partir de redes neurais construtivas do tipo KBANN. SBRN'98 – Simpósio Brasileiro de Redes Neurais. Belo Horizonte, 1998.

[Osório 99] OSORIO, F. S. Bibliografia – SHNS: Sistemas Híbridos Neuro-Simbólicos. Material disponível na Internet. Web: <http://www.inf.unisinos.br/~osorio/shns-refs.html>

[Osório 99a] OSORIO, F. S. e AMY, Bernard. *INSS : A Hybrid System for Constructive Machine Learning*. Neurocomputing, 26 (1999). Elsevier Publishing. (à ser publicado).

[Patel 97] PATEL, M. & HONAVAR, V. (Eds.) *Advances in Evolutionary Synthesis of Neural Systems*. Boston, MA - U.S.A. MIT Press, 1997.

[Porto 97] PORTO, V. W. *Neural-Evolutionary Systems*. In: Handbook of Neural Computation (section D2). E. Fiesler and R. Beale (Eds.) Institute of Physics and Oxford University Press. New York, NY - U.S.A., 1997. Web: <http://www.idiap.ch/publications/fiesler-96.1.bib.abs.html> or http://www.oup-usa.org/acadref/nc_accs.html

[Posey 92] POSEY, C.; KANDEL, A.; LANGHOLZ, G. *Fuzzy Hybrid Systems*. In: Hybrid Architectures for Intelligent Systems. A. Kandel and G. Langholz (Eds.). CRC Press, Boca Raton - Florida, 1992.

- [Potter 92] POTTER, Michell. *A Genetic Cascade-Correlation Learning Algorithm*. In: Proceedings of the COGANN-92 International Workshop on Combinations of Genetic Algorithms and Neural Networks, pp.123-133, IEEE Computer Society Press, 1992.
- [Rocha 92] ROCHA, A.F. & YAGER, R.R. *Neural Nets and Fuzzy Logic*. In: Hybrid Architectures for Intelligent Systems. A. Kandel and G. Langholz (Eds.). CRC Press, Boca Raton - Florida, 1992.
- [Sahami 95] SAHAMI, Mehran. *Generating Neural Networks Through the Induction of Threshold Logic Uni Trees*. In: Proceedings of the First International IEEE Symposium on Intelligence in Neural and Biological Systems. Washington, DC. May 1995.
- [Salomon 96] SALOMON, Ralf & VAN HEMMEN, Leo. Accelerating Backpropagation through Dynamic Self-Adaptation. *Neural Networks*, vol.9(4), pp.589-601, Pergamon Press. 1996.
- [Samad 92] SAMAD, T. *Hybrid Distributed/Local Connectionist Architectures*. In: Hybrid Architectures For Intelligent Systems, A. Kandel and G. Langholz (Eds.), Boca Raton - CRC Press. 1992.
- [Stuckelberg 96] STÜCKELBERG, M. & HILARIO, M. *Declarative Heuristics for Neural Network Design*. In: Proceedings of the ECAI'96 Workshop - Neural Networks & Structured Knowledge. Budapest, Hungary, August 1996.
- [Sulzberger 93] SULZBERGER, S.; Tschichold-Gurman, N.; VESTLI, S. *FUN: Optimization of Fuzzy Rule Based Systems Using Neural Networks*. IEEE Conference on Neural Networks, San Francisco, U.S.A. April 1993.
Web: <http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/fuzzy/systems/flie/>
- [Sun 95] SUN, Ron. (Ed.) *Integrating Rules and Connectionism for Robust Commonsense Reasoning*. John Wiley and Sons, 1995.
- [Sun 95a] SUN, Ron. *Robust Reasoning: Integrating Rule-Based and Similarity-Based Reasoning*. Artificial Intelligence Journal. June 1995. Web: <http://www.cs.ua.edu/sun/>
- [Towell 91] TOWELL, Geoffrey. *Symbolic Knowledge and Neural Networks: Insertion, Refinement and Extraction*. Ph.D. Thesis, Computer Science Dept., University of Wisconsin-Madison, U.S.A. 1991. Web: <http://www.cs.wisc.edu/~shavlik/uwml.html>
Ftp: [ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/\(towell.thesis.*.ps\)](ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/(towell.thesis.*.ps))
- [Towell 93] TOWELL, G. & SHAVLIK, J. *Extracting Refined Rules From Knowledge-Based Neural Nets*. Machine Learning, Kluwer Academic Publishers, Boston, pp.71-101, 13 (1993).
Web: <http://www.cs.wisc.edu/~shavlik/uwml.html> .
Ftp: <ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/towell.mlj93.ps>
- [Towell 94] TOWELL, G. & SHAVLIK, J. *Knowledge-Based Artificial Neural Networks*. Artificial Intelligence Journal, 70, pp.119-165. Elsevier Publishing, 1994.
Web: <http://www.cs.wisc.edu/~shavlik/uwml.html>
Ftp: <ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/towell.aij93.ps>
- [Utgoff 88] UTGOFF, Paul. *Perceptron Trees: A case study in hybrid concept representations*. In: Proceedings of the Seventh National Conference on Artificial Intelligence, St. Paul, MN. Morgan Kaufmann Publishing. 1988.

[Vestli 93] VESTLI, Sjur. *FLIE - Fuzzy-Logic Inference Engine*. Publié dans le WEB-Internet. Web: <http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/fuzzy/systems/flie/>
Ftp: <ftp://ftp.hiof.no/pub/Fuzzy/flie/> (fliefort.docu)

