Representação do Conhecimento em Sistemas Conexionistas: Tópicos em Análise

Pedro Paulo da Silva Ayrosa

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Profa.Sueli Bandeira T. Mendes, Ph. D.
(Presidente)

Profa.Doris Ferraz de Aragon, L.D.

Prof.Luis Alfredo V. de Carvalho, D.Sc.

RIO DE JANEIRO, RJ - BRASIL
SETEMBRO DE 1992

AYROSA, PEDRO PAULO DA SILVA

Representação do Conhecimento em Sistemas Conexionistas: Tópicos em Análise [Rio de Janeiro] 1992

XII, 119 p., 29.7 cm, (COPPE/UFRJ, M. Sc., ENGENHARIA DE SISTEMAS E COMPUTACÃO, 1992)

TESE - Universidade Federal do Rio de Janeiro, COPPE

1 - Redes Neuronais 2 - Representação do Conhecimento 3 - Conexionismo 4 - Álgebra Tensorial

I. COPPE/UFRJ II. Título(Série).

"Os padrões que os cientistas observam na natureza estão intimamente relacionados com os padrões das suas mentes, com os seus conceitos, pensamentos e valores. Por isso, os resultados científicos que obtêm e as implicações tecnológicas que investigam estarão condicionados pela estrutura de suas mentes. Embora grande parte de suas pesquisas detalhadas não seja explicitamente dependente dos seus sistemas de valores, a estrutura mais abrangente dentro da qual essas pesquisas são efetuadas nunca será independente de valores. Os cientistas, portanto, são responsáveis, não apenas intelectual mas também moralmente, por suas pesquisas."

Fritjot Capra

Aos meus pais

Agradecimentos

À Profa. Sueli Mendes, pela amizade, orientação e paciência no acompanhamento e desenvolvimento deste trabalho.

À Profa. Doris, pelo incentivo inicial, respeito e exemplo profissional.

Ao Prof. Luís Alfredo por suas aulas de redes neuronais sempre brilhantes.

Ao Prof. Paul Smolensky da *"University of Colorado at Boulder"* pelo fornecimento do material sobre representação por produto tensorial de sua autoria.

À Prof. Nanci Vieira e ao Prof. Roosevelt Dias pelas suas amizades.

Aos amigos Luiz Adauto e Cláudia pela convívio sempre fraterno.

Ao pessoal do ILTC onde tudo começou.

A minha família e amigos por inúmeros motivos.

Ao CNPQ, pelo apoio financeiro a este trabalho.

Resumo da Tese apresentada à COPPE como parte dos requisitos necessários para a obtencão do grau de Mestre em Ciências (M. Sc.)

Representação do Conhecimento em Sistemas Conexionistas:
Tópicos em Análise.

Pedro Paulo da Silva Ayrosa

Setembro de 1992

Orientador: Sueli Bandeira Teixeira Mendes
Programa: Engenharia de Sistemas e Computacão

Atualmente são desenvolvidas várias pesquisas em modelos conexionistas para manipulação de estruturas simbólicas, tais modelos são conhecidos como modelos conexionistas de alto nível. Uma aplicação destes modelos em Inteligência Artificial é a representação do conhecimento estruturado simbólico em redes neuronais. O objetivo deste trabalho é uma pequena análise dos principais conceitos e tendências desta nova área de pesquisa.

Abstract of Thesis presented to COPPE as partial fulfillment of the requirements for the degree of Master of Science (M. Sc.)

Knowledge Representation in Connectionist Systems:
Topics under Analysis.

Pedro Paulo da Silva Ayrosa

September, 1992

Thesis Supervisor: Sueli Bandeira Teixeira Mendes
Department: Programa de Engenharia de Sistemas e Computacão

Currently, several researchs have been developed about connectionistics models for handling symbolic structures. Such models are known as high level connectionistics models. A representation of symbolic structure knowledge in neural networks is a example of these models applications in Artifical Intelligence. The aim of this work is to offer a simple analysis of the main concepts and tendencies of this new area of research.

Índice

I Introdução 1

I.1 Considerações Iniciais 1

I.2 Contribuição deste Trabalho 3

I.3 Descrição dos Capítulos 5

II Modelos Conexionistas 7

II.1 A Evolução de uma Idéia 7

II.2 O Modelo Conexionista Básico 10

II.2.1 Características da Unidade de Processamento 13

II.2.2 Tipos de Regras de Aprendizado 14

II.2.3 Topologia da Rede 17

III Representação de Conhecimento em Sistemas Conexionistas 19

III.1 Introdução 19

III.2 Representação Puramente Local 20

III.2.1 "Fuzzy Cognitive Map" 21

III.3 Representação Parcialmente Distribuída 26

III.3.1 'Coarse Coding' 29

III.3.2 A Questão das Micro-características 32

III.3.3 Esquemas Conexionista 32

III.4 Representação Holográfica 39

IV Processamento Simbólico Conexionista 41

IV.1 Modelos Conexionistas de Alto Nível 41

IV.1.1 Características dos Modelos Conexionistas e Simbolistas . . . 42

IV.2 Recirculação Simbólica (*"Symbol Recirculation"*) 44

IV.3 A Questão da Modularidade 45

IV.4 Algumas Propostas para o Processamento Simbólico Conexionista . . 47

IV.4.1 Representação de hierarquias via descrições reduzidas 47

IV.4.2 Memória Auto-Associativa Recursiva - *"RAAM"* 48

V Sistema de Produção Conexionista Distribuido 50

V.1 Introdução 50

V.2 Sistema de Produção Clássico 50

V.2.1 Estrutura de um Sistema de Produção 51

V.2.2 Estratégia de Controle nos *SP's* 53

V.3 SP Conexionista Distribuído - *(DCPS)* 53

V.4 A Arquitetura do Interpretador do *DCPS* 57

V.4.1 Memória de Trabalho 57

V.4.2 Espaço de Regras 58

V.4.3 Espaços de Cláusulas 59

V.4.4 Espaço de Ligação 61

VI Representação por Produto Tensorial **63**

VI.1 Introdução 63

VI.1.1 O que é Representação por Produto Tensorial 63

VI.1.2 Características da Representação por Produto Tensorial 65

VI.2 Álgebra Tensorial 66

VI.3 Definição de Produto Tensorial relativo a uma Base 67

VI.4 Definição de Produto Tensorial independente de uma Base 72

VI.4.1 Espaço Dual 72

VI.4.2 Produto Tensorial de Funcionais 77

VI.4.3 Produto Tensorial de Vetores 78

VI.5 Representação Conexionista por Produto Tensorial 81

VI.5.1 Representação Conexionista 81

VI.5.2 Decomposição de Estruturas Simbólicas 82

VI.5.3 Representação Conexionista de Conjunções 86

VI.5.4 Representação Conexionista da Ligação de Variáveis 88

VI.5.5 Representação por Produto Tensorial 89

VI.6 Tipos de Representações da RTP 90

VI.7 "Unbinding" 90

VI.8 Estudo de um Exemplo: "Frames" 93

VI.8.1 O modelo básico 93

VI.8.2 A RTP do "FRAME" Ferir 94

VII Conclusões 102

A Implementações 113

A.1 Implementação da Representação Puramente Local 113

Lista de Figuras

I.1 Taxonomia dos Sistemas Conexionista 4

II.1 Modelo Geral de uma Unidade u_i 13

III.1 Esquema dos futuros sistemas híbridos 22

III.2 Esquema das relações causais 24

III.3 Tipos de representação: puramente local, parcialmente distribuída e holográfica 27

III.4 Forma simples de codificar um ponto no espaço bidimensional utilizando-se dois grupos de unidades 29

III.5 As quatros possibilidades para a representação de dois pontos 29

III.6 Esquema da rede *"Brain-State-in-the-Box"* 37

V.1 Esquema dos diversos espaços do DCPS 55

V.2 Rede "winner-take-all" composta de dois "cliques" com três unidades, onde cada um requer $\frac{(N^2-N)}{2}$ conexões 59

V.3 Espaço de unidades com $\frac{(N^2-N)}{2}$ conexões bidirecionais 60

V.4 Unidade Reguladora 61

VI.1 Definição de produto tensorial independente de uma base 68

Lista de Tabelas

III.1 Tabela de padrões a serem aprendidos 35

III.2 Tabela de experimentos para verificação de valores default 39

III.3 Tabela de valores default 39

V.1 Tabela de Campo Receptivo 58

VI.1 Tabela de símbolos e vetores característica 94

VI.2 Representação do conceito "João feriu Maria" 94

VI.3 Vetores associados ao conceito "João feriu Maria" 95

Capítulo I

Introdução

I.1 Considerações Iniciais

A década de 90 será, provavelmente, marcada pelo desenvolvimento da biocomputação, isto é, criação/desenvolvimento de hardware e software com inspiração em sistemas biológicos. O conjunto de teorias e técnicas englobadas pelo termo biocomputação são: neurocomputação ou neoconexionismo, algoritmos genéticos, *'iterated function systems'*, *'L-systems'*, etc.

A idéia que sustenta este paradigma computacional pode ser resumida com a seguinte afirmação: é possível a construção de sistemas extremamente complexos com alto grau de redundância, tolerância à falha e comportamento adaptativo a partir de um pequeno conjunto de regras, conforme é observável nos bioorganismos [VALD91].

Assim, dentro do mega-paradigma da biocomputação existe uma série de micro-paradigmas que estão sendo estudados e desenvolvidos dentre os quais a neurocomputação ou o neoconexionismo é o que apresenta maior possibilidade de aplicação em problemas reais a curtíssimo espaço de tempo.

Uma linha de pesquisa que tenta combinar o processamento simbólico tradicional com as novas técnicas não simbólicas é o **processamento conexionista de alto nível** ou **processamento simbólico conexionista.**

Recentemente, a construção de sistemas conexionistas para proces-

samento simbólico tem despertado o interesse da comunidade internacional de Inteligência Artificial. Um periódico tradicional (*Artificial Intelligence - An International Journal*) dedicou dois números inteiros (*Vol.46(1-2)*) à questão do Processamento Simbólico Conexionista. Conforme discutiremos neste trabalho , isto aponta para a tendência do desenvolvimento de uma classe de sistemas híbridos que exploram várias características dos modelos paralelo-distribuídos e dos sistemas simbólicos tradicionais.

Um outro ramo de pesquisa que não está isolado desta problemática (integração do processamento simbólico com o conexionista) é o processamento conexionista de linguagem natural (PCLN). Assim, os estudos em PCLN têm estabelecido novas formas de representação que não possuem paralelo com as representações tradicionais da IA. Apesar de ainda ser difícil estabelecer limites precisos dos tipos de representação conexionista - já que muitos modelos utilizam varias formas de representações e combinações destas - tentaremos analisar os principais conceitos, características, vantagens e desvantagens das representações conexionistas do conhecimento, tendo em vista os modelos conexionistas para processamento simbólico.

Uma forma de classificar a evolução dos computadores é dividí-los em gerações conforme os seguintes parâmetros: tecnologia de construção, tecnologia circuital, tecnologia programacional,arquitetura, etc.Desta forma, temos até agora 5 gerações estabelecidas . A quinta geração é a que apresenta um interesse especial para a inteligência artificial, veja o artigo de [ACAR91] para uma análise dos objetivos, desenvolvimento e implicações da quinta geração de computadores.

A Sexta-Geração, apesar de já esta sendo esboçada desde o final dos anos 80, será oficialmente lançada pelo governo do Japão em 1992 . Seguindo o modelo do projeto anterior (Quinta-Geração), o novo projeto a ser lançado pelo o Ministério da Industria e do Comércio Internacional do Japão (MITI) prevê um custo de 200 bilhões de *yens* e envolverá universidades, laboratórios governamentais, empresas privadas de todo o Japão e contribuições de toda a comunidade científica internacional, com duração de 10 anos [CROSS91].

Segundo [SOUC88] o Projeto da Sexta-Geração está dividido em três

partes: *Ciência, Tecnologia e Aplicações.* O conjunto de pesquisas sobre o rótulo de ciência tenta responder às seguintes perguntas:

- O que é inteligência, mente e cognição?
- Como pensamos, vemos e ouvimos?
- Qual é a arquitetura do cérebro humano?
- O que controla o comportamento?
- Como opera a memória?
- etc.

A partir do conhecimento produzido na etapa anterior (ciência) é desenvolvido um conjunto de técnicas para o aprendizado por máquina, tratamento de estruturas simbólicas por arquiteturas não-convencionais, reconhecimento de padrões complexos, etc. Como resultado das duas etapas anteriores (ciência e tecnologia) temos a etapa das *aplicações*: sistemas especialistas híbridos, instrumentos de controle inteligente, sistemas CAD/CAM/CAE inteligentes,robôs com comportamento mais próximo dos organismos vivos, etc.

Para um estudo mais detalhado desta nova geração de computadores veja [SOUC88], [SOUC89].

Como vemos, o tratamento simbolico por sistemas conexionista ainda se apresenta em fase de gestação e é apenas um tópico do conjunto de conhecimento em processo de desenvolvimento dos sistemas da sexta-geração. Dentro da taxonomia apresentada por [SOUC89] os sistemas conexionistas simbólicos e estudos correlatos se enquadram no nível mais abstrato denominado *coarse grain intelligence* veja figura I.1.

I.2 Contribuição deste Trabalho

A contribuição deste trabalho é tentar reunir vários conceitos dispersos na literatura sob uma única terminologia e em um só lugar. Servindo de referência para futuros

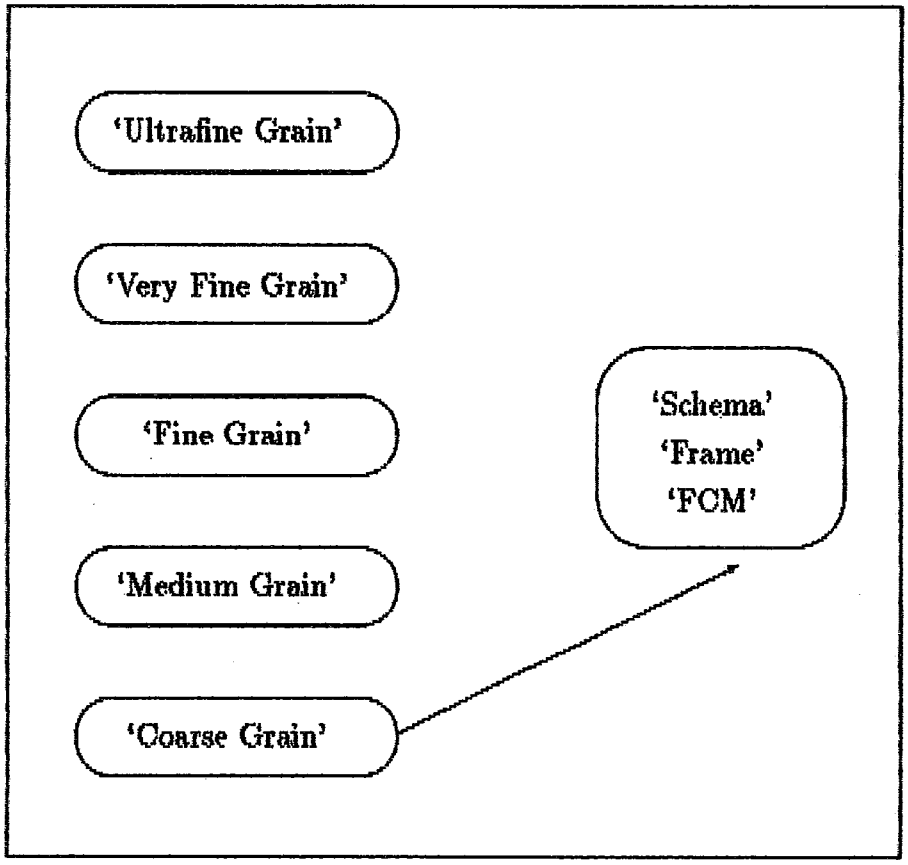

Figura I.1: Taxonomia dos Sistemas Conexionista

trabalhos que utilizem uma representação conexionista ou um tratamento simbólico através do paradigma conexionista.

I.3 Descrição dos Capítulos

No Capítulo II tratamos de forma introdutória - já que temos ótimos trabalhos neste campo - os princípios dos modelos conexionista. Mostramos a evolução das idéias a respeito do processamento neuronal e descrevemos o modelo conexionista básico sobre três aspectos: características da unidade de processamento, tipos de regras de aprendizado e topologia das redes neuronais artificiais.

O Capítulo III aborda a representação do conhecimento em sistemas conexionistas, mais precisamente, as representações ditas semânticas. Analisamos os três tipos básicos de representação - puramente local (PL), parcialmente distribuída (PD) e totalmente distribuída (TD) - mostrando através de exemplos - no caso da representaçào PL e PD - as vantagens e desvantagens das referidas representações.

O Capítulo IV procura explicitar as razões que norteiam a pesquisa de modelos conexionistas de alto-nível descrevendo técnicas atuais que objetivam a construção de sistemas híbridos com uma micro-semântica , característica das redes neuronais, e uma macro-semântica, característica dos sistemas simbólicos tradicionais.

O Capítulo V estuda o sistema de produção conexionista distribuído mostrando uma das tentativas da união do processamento simbólico com o processamento conexionista. Mostramos a estrutura do sistema e como certos conceitos da representação conexionita foram usados para a representação simbólica.

O Capítulo VI é o que apresenta uma das teorias mais modernas de formalização da representação conexionista conhecida como Representação por Produto Tensorial. Estudamos os principais resultados da álgebra tensorial e suas aplicações neste tipo de representação. Ressaltamos também que diversos pontos enfocados nos capítulos anteriores são englobados por esta teoria.

No Capítulo VII, são feitas considerações gerais sobre os tipos de representações estudadas e as conclusões do trabalho, bem como, direções de pesquisas possíveis para futuros trabalhos.

Além disso, os Apêndices contém todos os arquivos fontes e parâmetros para a reprodução das simulações realizadas neste trabalho.

Capítulo II

Modelos Conexionistas

II.1 A Evolução de uma Idéia

Os modelos conexionistas são modelos que apresentam características bem diferentes dos modelos tradicionais para computação . Enquanto os modelos tradicionais, chamados simbolistas, encaram o comportamento inteligente como um sistema que pode ser descrito com processos de entrada, armazenamento, recuperação e transformação de informações, os modelos conexionistas encaram o comportamento inteligente como resultado da interação de diversos elementos computacionais simples, altamente interconectados, inspirados nos neurônios biológicos.

A visão simbolista, também chamada cognitiva, foi "codificada" no artigo clássico intitulado *"Computer Science as Empirical Inquiring: Symbols and Search"* de Newell e Simon [NEWE76] onde as principais idéias são formuladas tendo como base os seguintes conceitos [COST86]:

Símbolos: são formas, "padrões", estampados em um substrato físico;

Estrutura de Símbolos: conjunto de instâncias de símbolos, relacionadas entre si de algum modo físico;

Sistema Físico de Símbolos: máquina capaz de executar processos que, através do tempo, produzem uma sucessão de estruturas simbólicas, derivadas umas das outras;

Hipótese do Sistema Físico de Símbolos: Um sistema físico de símbolos tem os meios necessários e suficientes para a ação inteligente geral.

A partir da *Hipótese do Sistema Físico de Símbolos* ficam claros os pressupostos assumidos pela ciência cognitiva, tais pressupostos, entretanto, são amplamente questionados por muitos conexionistas, como por exemplo :

- "A informação e o processamento de informação podem ser estudados como símbolos de acordo com determinadas regras (algorítmos ou procedimentos formais)." [COST86]
- A *não* necessidade de referências às estruturas biológicas e físicas para o estudo das funções cognitivas.

Assim, o conexionismo se apresenta, inicialmente, como uma linha de estudo que rompe certas posições clássicas no estudo da cognição humana. Esta posição de divergência pode ser resumida na seguinte comparação [CARV89b]:

- Cognitivistas
  - O cérebro é um processador de símbolos;
  - A inteligência se materializa em heurísticas;
  - A lógica é uma ferramenta adequada para descrever e especificar processos cognitivos;
  - Os processos mentais são seqüenciais e centralizados.
- Conexionistas
  - O cérebro é uma rede de processadores simples que trocam sinais entre si, cooperando ou competindo;
  - A inteligência está nas conexões entre os processadores;
  - Os processos cognitivos devem ser descritos e especificados utilizando conceitos da neurociência e do processamento paralelo/distribuído;

Estas duas maneiras de se encarar a cognição e, consequentemente, o papel dos computadores na realização de tarefas ditas inteligente, tem gerado muita polêmica.Um exemplo que não pode ser negligenciado é o ataque de Fodor e Pylyshyn [FODO88] à questão do tratamento do conhecimento estruturado e ao uso de regras por sistemas conexionistas.

Apesar deste debate ser extremamente interessante e ter apenas começado, não faz parte deste trabalho uma abordagem detalhada destas questões . Mas um ponto que devemos ressaltar é o aprimoramento das técnicas e dos modelos que constituem contra-exemplos às argumentações iniciais [TOUR88], [TOUR90], [SMOL87], [SMOL88], [SMOL90], [LEGE90].

Historicamente, a origem do interesse por sistemas conexionista, mais propriamente redes neuronais, está na neurociência. Podemos acompanhar a evolução das idéias nesta área dividindo a história do paradigma conexionista em 4 períodos [EBER90].

O primeiro período (1890-1969) teve início com os trabalhos de William James, onde pela primeira vez foi publicado um conjunto de fatos relacionados com a estrutura do cérebro e sua função, tal como o princípio da memória associativa (1890). Mais de meio século depois (1943), Warren McCulloch e Walter Pitts publicam um dos mais famosos trabalhos nesta área, onde descrevem o primeiro modelo matemático de uma rede neuronal artificial. Neste trabalho, os autores desmonstram teoremas que explicitam certas propriedades compatíveis com o conhecimento neuro-biológico da época (década de 40). A seguir, temos a contribuição de Donald Hebb que formula, de maneira especulativa, como se desenvolve o aprendizado à nível celular.Em 1950 com o trabalho do neurofisilogista Karl Lashley, temos a atenção voltada para a representação da informação nos sistema biológicos, e a questão da representação localizada *versus* distribuída é levantada. Frank Rosenblatt publica (1958) um trabalho ,também clássico, onde define uma rede neuronal chamada perceptron que é uma generalização do modelo de McCulloch-Pitts incorporando um procedimento de aprendizado. Neste trabalho, encontramos a raiz do sistemas hoje conhecidos como *"winner-take-all"*. Este período termina com a publicação do livro *Perceptrons* (1969) de Minsky e Papert. Este trabalho demonstrou ser os sistemas,

até então estudados, extremamente limitados, isto é, só tratavam, satisfatoriamente, de problemas cujas soluções fossem linearmente separáveis.

Após este período inicial, temos um período de aparente estagnação (1969-1982) no desenvolvimento da pesquisa em sistemas conexionistas. Porém, pesquisadores dentre os quais James Anderson, Stephem Grossberg e Teuvo Kohonen continuaram desenvolvendo e publicando trabalhos sem, contudo, despertar o interesse da comunidade científica internacional.

A partir da publicação dos trabalhos de Hopfield (1982) estabelecendo um modelo conexionista com propriedades interessantes, como estabilidade da rede sob determinadas condições, por exemplo, é que temos um incremento de pesquisas que culminarão com a publicação do livro *Parallel Distribued Processing* por D.E. Rumelhard e J.L. McClelland (1986). Este período de renascimento do conexionismo gerou grande interesse na comunidade científica em muito pouco tempo. Basta observar o números crescente de congressos, simpósios e encontros que tratam do tema.

O último período que começou em 1987, denominado de neoconexionismo, é caracterizado pelo desenvolvimento de ferramentas e simuladores para computadores pessoais que permitem a exploração de diversos modelos em diversos domínios. Assim, temos várias aplicações tais como: previsão do comportamento do mercado financeiro, análise de resultados gerados em exames biomédicos, composição musical, etc. Também temos diversos produtos que tentam combinar as características dos modelos conexionistas com sistemas especialistas e banco de dados, apontando, assim, para uma integração dos modelos conexionistas com os sistemas convencionais.

II.2 O Modelo Conexionista Básico

Apesar de termos vários textos que tratam de maneira introdutória o estudo dos sistemas conexionistas [KOHO87],[LIPP87],[WASS89],[SOUC88],[SIMP90] faremos, aqui, uma breve revisão dos principais conceitos, a fim de unificar os diversos termos

e explicitar a maneira como eles se interrelacionam.

Uma dificuldade inicial é estabelecer uma definição precisa e ampla dos sistemas conexionistas. Assim como o termo *inteligência artificial* apresenta mais de 150 definições [CLEI91], o termo *sistema conexionista* também apresenta diversas definições e sinônimos, tais como : redes neuronais artificiais (RNA), sistemas paralelo distribuído (PDP), redes adaptativas (RA), etc.Uma das principais razões para a diversidade de definições e termos é o fato dos sistemas conexionistas sofrerem influências direta de várias disciplinas tais como: matemática, psicologia, neurociência, física, engenharia, filosofia, ciência da computação, biologia, etc.Observamos, assim, diversos e diferentes termos para expressar um mesmo fato.

Por exemplo, nas três definições abaixo temos diferentes definições para o termo sistemas conexionistas (redes neuronais).Cada uma delas expressa de maneira genérica a visão ou enfoque dos seus autores, mostrando, assim, que ainda estamos em fase inicial de pesquisa e estruturação desta disciplina [FERR89].

Definição 1: *"Redes neuronais artificiais são circuitos elétricos ou óticos projetados como metáforas para o sistema computacional de neurobiologia. Diferem dos VLSI mais convencionais pela ênfase na grande conectividade dos dispositivos ativos; uso de propriedades de dispositivos analógicos em computação; uso da evolução no tempo e atrasos no tempo como um sistema computacional dinâmico, em lugar de apenas como uma transição entre estados lógicos; uso de realimentação na estrutura computacional; e o uso extensivo de aprendizado em nível de hardware"*,[HOPF88]

Definição 2: *"Redes neuronais: um sistema dinâmico com a topologia de um grafo orientado, que pode levar a cabo o processamento de informações por meio de sua resposta a entradas contínuas ou episódicas. Os nodos nas redes neuronais são chamados elementos de processamento, os arcos orientados (os canais de informação) são chamados interconectores. Cada elemento de processamento é dotado com alguma memória local. O processamento que ocorre em cada elemento de processamento (este processamento é definido pela função de transferência do elemento de processamento) deve depender somente dos valores na*

memória local. Os elementos de processamento operam ou continuamente ou são atualizados de acordo com uma função pré-estabelecida. Tipicamente as redes neuronais são compostas de conjuntos de elementos de processamento, denominados camadas, nos quais todos elementos de processamento possuem a mesma função de transferência.",[SOUC88]

Definição 3: *"Um modelo conexionista é uma estrutura de processamento de informação distribuída e paralela. Ela é formada por unidades de processamento, comumente chamada de nós, neurônios ou células, interconectadas por arcos unidirecionais, também chamados de ligações, conexões ou sinapses. Os nós possuem memória local e podem realizar operações de processamento de informação localizada. Cada célula possui uma única saída (axônio), a qual pode se ramificar em muitas ligações colaterais (cada ramificação possuindo o mesmo sinal da saída do neurônio). Formalmente, o sinal de saída do nó pode ser equacionado de diversas maneiras. Todo o processamento que se realiza em cada unidade deve ser completamente local, isto é, deve depender apenas dos valores correntes dos sinais de entrada que chegam dos neurônio através das conexões. Estes valores atuam sobre os valores armazenados na memória local da célula.*",[HECH88]

Rumelhard et al. [RUME86] estabeleceram um *"framework"* que tem sido reproduzido em diversos trabalhos [PESS89a],[PESS90a],[CARD90],[DENI91]. Podemos considerar esta abordagem como um enfoque *"bottom-up"* onde são definidos os diversos conceitos que formam um modelo conexionista. Optamos, neste trabalho, por uma abordagem *"top-down"*, pois a consideramos uma forma mais adequada para apresentação de uma síntese sobre o tema.

Assim, um modelo conexionista pode ser visto como formado de três grupos de definições ou parâmetros:

- Características da Unidade de Processamento,
- Tipo de Regra de Aprendizado,
- Topologia da Rede.

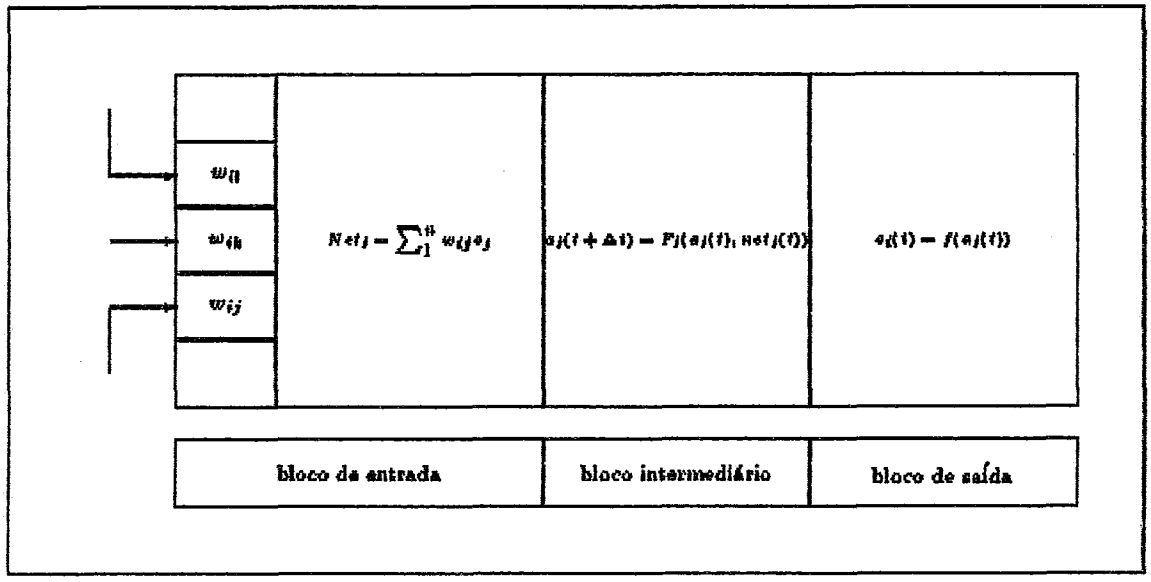

Figura II.1: Modelo Geral de uma Unidade u_i

II.2.1 Características da Unidade de Processamento

Como vimos anteriormente, a unidade de processamento é a célula básica dos modelos conexionistas. Ela corresponde a um modelo simplificado do neurônio biológico. Em [CARV89a] temos detalhes do funcionamento dos neurônios biológicos e uma analogia funcional com as unidades de processamento dos sistemas neuronais artificiais.

Uma unidade de processamento u_i é formada por três blocos lógicos (veja figura II.1): *bloco de entrada* responsável pela reunião dos sinais de entrada vindos de outras unidades ou do ambiente, *bloco intermediário* responsável pelo *estado de ativação* da unidade em determinado instante t e, finalmente, um *bloco de saída* responsável pelo valor de saída da unidade no instante t. Vejamos, agora, cada um destes blocos isoladamente.

O *bloco de entrada* combina os valores provenientes da saída de diversas fontes expresso por um vetor de entrada $O = (o_1, ..., o_j, ..., o_n)$, com os respectivos pesos das ligações. Cada peso é expresso por um valor (variável) w_{ik}, isto é, o valor da conexão da saída o_k à entrada da unidade u_i.O conjunto de todos os pesos entre todas as ligações (*padrão de conexão*) das unidades de uma rede é expresso por uma matriz bidimensional, chamada *matriz de conexão*.Em resumo, este bloco combinará

todas os sinais de entrada com seus respectivos pesos, fornecendo um único sinal de entrada denominado net_i, resultado da aplicação de uma *regra de propagação*, onde:

$$Net_i = \sum_1^n w_{ij} o_j$$

Após a obtenção do valor net_i pela aplicação da *regra de propagação* temos que obter outro valor, denominado *estado de ativação* $a_i(t)$. Isto é realizado no bloco chamado *bloco intermediário*, onde é utilizado uma função F denominada *regra de ativação* que combina o valor corrente $a_i(t)$ com o net_i obtendo assim o novo valor de $a_i(t)$:

$$a_i(t + \Delta t) = F_i(a_i(t), net_i(t))$$

O terceiro e último bloco (*bloco de saída*) é responsável por transmitir um valor o_i (*valor de saída*) da unidade em questão para todas as outras unidades a ela interconectada. O *valor de saída* de cada u_i é determinado pelo *estado de ativação* $a_i(t)$ e uma função f denominada *função de saída*:

$$o_i(t) = f(a_i(t))$$

II.2.2 Tipos de Regras de Aprendizado

As *regras de aprendizado* têm por objetivo modificar o *padrão de conexão* da rede em função do aprendizado ou adaptação. Algumas regras utilizadas são:

Hebb: Esta regra estabelece que o ajuste depende, somente, da saída desejada e das entradas da unidade de processamento. A equação utilizada para descrevê-la é:

$$w_{ij}(t + 1) = w_{ij}(t) + \eta a_i(t) a_j(t)$$

ou em notação simplificada:

$$\Delta w_{ij} = \eta a_i a_j,$$

onde η é uma constante positiva denominada *taxa de aprendizado*. A equação descreve que: se a saída desejada e a entrada estão, ambas, acima do limiar, então ocorrerá um incremento nos pesos determinado pela taxa de aprendizado.

Anti-Hebb: Como a equação que descreve a regra de Hebb só permite o incremento dos pesos, costuma-se, por diversos motivos, utilizar regras denominadas Anti-Hebbianas. Estas regras incorporam um termo de decaimento responsável pelo decremento dos pesos. Uma das equações comumente utilizada é a equação diferencial:

$$\dot{w_{ij}} = -Aw_{ij} + a_i a_j$$

Diferencial-Hebb: Regra utilizada por [KOSK85], [KOSK91] é descrita pela equação:

$$\dot{w_{ij}} = -w_{ij} + \dot{S}(a_i)\dot{S}(a_j),$$

onde S é a função sigmóide.

Widrow-Hoff: Considerando que a saída de uma unidade é dada pela equação :

$$y_j = w_{0j} + \sum_{i-1}^{n} w_{ij}\dot{x}_i$$

a regra de aprendizado associada é descrita por:

$$w_{ij}(t + \Delta t) = w_{ij}(t) + \lambda \epsilon_j(t) x_i(t)$$

onde ϵ_j é calculado pela subtração do valor real do valor desejado ($\epsilon_j = d_j - y_j$), e λ é o fator de proporcionalidade.

Para um estudo detalhado das diversas regras de aprendizado tais como: ADALINE, perceptron, backpropagation, Instar, Outstar,Kosko/Klopf, Koronen, Hopfield, veja [MURP91], [SIMP90] e [WASS89].

Em função da regra escolhida , do funcionamento da rede e do nível de supervisionamento empregado, podemos ter diversas classificações como exposto em [PESS90a]:

- Quanto ao Paradigma de Funcionamento:

  Auto-Associador: Modelos em que um conjunto de padrões são armazenados para um posterior recuperação através de atributos particulares dos padrões armazenados. Os modelos associativos mais comuns são: rede de Hopfied, BSB, ART. Uma aplicação possível para estes modelos é utilizá-los como memória associativa onde dado um fragmento de um padrão o sistema é capaz de recuperar o padrão completo.

  Associador de Padrões: Modelos em que um conjunto de padrões é mapeado em um outro conjunto de padrões.

  Classificador: Modelos em que um conjunto de padrões é associado a um outro grupo de padrões agrupados em categorias.

  Detetor de Regularidades: Modelos que são capazes de categorizar um conjunto de padrões de entrada, isto é, o modelo não possui um conjunto de classes determinado a priori, assim, ele deve determinar as categorias relevantes baseado nas regularidades estatísticas dos padrões.

- Quanto ao Nível de Supervisionamento:

  Aprendizado Supervisionado: Neste tipo de aprendizado é fornecido à rede um conjunto de pares onde cada par é composto do padrão de entrada e do padrão desejado na saída,para serem comparados e haver a correção.

  Aprendizado por Reforço: Este tipo de aprendizado requer um padrão de entrada e um sinal externo que pode ser interpretado como uma medida de adequação das suas ações recentes.

  Aprendizado Não-Supervisionado: A rede recebe somente o padrão de entrada e o classifica internamente, ressaltando por si só as propriedades interessantes dos estímulos de entrada.

- Quanto ao Tipo de Regras:

  Regras de Correlação: São caracterizadas por ter como único meio de se alterar os pesos das conexões, as ativações das unidades.

Regras de Correção de Erros: Neste tipo de regras a rede utiliza a resposta a um determinado padrão de entrada e compara este resultado com o valor desejado. Caso ocorra uma discrepância entre o valor obtido e o desejado esta diferença é utilizada na alteração dos pesos.

II.2.3 Topologia da Rede

O terceiro e último grupo de definições corresponde à *topologia da rede*. Atualmente a organização do conhecimento sobre redes neuronais tem tido como base uma taxonomia que toma como primeiro ponto de referência a topologia . Este fato é consequência da observação que redes com estruturas (topologias) similares frequentemente possuem atividades, regras de aprendizado e aplicações também similares.

Devemos, assim, entender como topologia os diversos parâmetros associados à estrutura de uma rede neuronal. Os parâmetros comumente utilizados para expressar a topologia de uma rede são:

- número de camadas
- número de unidades por camada
- tipos de interconexão
- direção do fluxo de informação

Considerando que uma rede neuronal pode ser dividida estruturalmente em três níveis (micro,meso,macro), a topologia corresponde ao nível intermediário (meso) da especificação da rede . O nível classificado como micro-estrutural corresponde às especificações das unidades e macro-estrutural abrange todos as especificações que permitem interligar diferentes redes para resolver problemas não tratáveis, satisfatoriamente, por um único modelo. Desta forma, podemos distinguir 6 tipos básicos de topologias de redes neuronais:

topologia multicamada: esta estrutura está normalmente associada a uma dinâmica recorrente e sua aplicação típica é como autoassociador (ex. perceptron).

topologia de camada única: como a estrutura anterior, esta topologia está associada a uma dinâmica recorrente e as aplicações típicas são autoassociativa e em otimização (ex. Hopfield).

topologia topográfica: corresponde aos modelos propostos por [KOHO87] e as aplicações mais comuns são: compressão de dados, otimização,etc.

topologia de camada dupla com realimentação: apresenta uma dinâmica recorrente com aplicações heteroassociativas e casamento de padrões (ex. memória associativa bidirecional).

topologia multicamada/em rede cooperativa/competitiva: dinâmica competitiva/cooperativa com aplicações em reconhecimento de padrões espaciais, temporais e espaço-temporais (ex. neocognitron).

topologia híbrida: resultado da combinação de diversos tipos de estruturas primárias classificadas anteriormente.

Capítulo III

Representação de Conhecimento em Sistemas Conexionistas

III.1 Introdução

O que a princípio parecia caracterizar um conflito entre os paradigmas simbolista e conexionista agora parece caminhar para uma síntese, onde as principais características de cada paradigma moldam os alicerces dos futuros sistemas híbridos [MINS91].

Estes dois paradigmas mais divulgados na comunidades da Ciência da Computação e ,especialmente, na comunidade de Inteligência Artificial, na realidade são os dois extremos de diversos outros paradigmas [WEST91]. De alguma maneira estas diferentes formas de se abordar um problema terão que ser combinadas, segundo diversos parâmetros, para uma fundamentação teórica e ampla que forneça caminhos para a problemática da Inteligência Artificial.

Para certas aplicações de Sistemas Conexionistas em Inteligência Artificial precisamos que a rede incorpore algum conhecimento sobre o problema particular bem como um método de manipulação deste conhecimento.

Neste capítulo abordaremos as três formas primárias de se representar o conhecimento nos sistemas conexionistas :

- *Representação Puramente Local*

- *Representação Parcialmente Distribuída*
- *Representação Totalmente Distribuída ou Holográfrica*

Uma questão que se apresenta é o que entendemos por conhecimento. Não estamos aqui interessados em abordagens epistemológicas e nas limitações resultantes da aceitação de algumas hipóteses vinculadas por uma das diversas escolas filosóficas. Porém, temos que assumir uma hipotése de trabalho, isto é, o que significa o termo *representação do conhecimento.*

Adotaremos o seguinte conceito : *representação do conhecimento é toda linguagem capaz de representar uma crença verdadeira justificada* [CARD90].

Considerando o conceito expresso acima temos que a representação do conhecimento está profundamente relacionada à estrutura que armazena este conhecimento, isto é, o processo de armazenagem e representação do conhecimento reflete o próprio tipo de conhecimento, e de maneira análoga, o tipo de conhecimento direciona, de certa maneira, o processo de armazenagem e representação. Dentro deste contexto, as formas de representação do conhecimento em sistemas conexionistas apresentam características e propriedades interessantes que ainda estão em fase inicial de estudo e que talvez estabeleçam formas mais eficientes de resolução ,por máquinas, de velhos problemas.

III.2 Representação Puramente Local

Não trataremos aqui da evolução das idéias sobre a representação puramente local, que foram resultantes do desenvolvimento da neurofisiologia [PESS89a], mas sim dos conceitos gerais e aplicações deste esquena de representação.

Como o nome sugere, a representação puramente local estabelece o esquema de memória em que cada unidade u_i representa um, e somente um, conceito, objeto ,relação ou regra. Este é considerado o esquema mais simples de representação do conhecimento em sistemas conexionista e apresenta a grande desvantagem da baixa tolerância a falhas. Assim, se por algum motivo uma ou várias

unidades apresentarem um funcionamento deficiente, a rede sofre profundamente o efeito deste problema, apresentado um desempenho pobre e, até mesmo, entrando em colapso total. Mesmo assim, este esquema apresenta a vantagem da simplicidade da implementação e de tornar inteligível, de certa maneira, o conjunto e estrutura do conhecimento representado.

III.2.1 "Fuzzy Cognitive Map"

Vejamos, nesta seção, um modelo que utiliza a representação puramente local para descrever um domínio, mostrando de forma clara as vantagens e desvantagens deste esquema de representação conexionista.

O conceito de *"Fuzzy Cognitive Map"* - *FCM* foi desenvolvido por [KOSK85] e tem tido inúmeras aplicações [TABE87], [STYB88], [ZHAN88].

Este modelo tem sido interessante por ampliar os domínios de aplicação dos sistemas especialista tradicionais. Na realidade alguns autores já o consideram como protótipos dos futuros sistemas especialistas híbridos, onde sistemas serão formados por 4 módulos básicos:

- sistema de regras se-então-senão,
- algoritmos genéticos,
- redes neuronais e
- sistemas *fuzzy*,

que trabalharão de maneira integrada e cooperativamente na solução de problemas (veja figura III.1). Nas aplicações atuais o modelo FCM se apresenta como produto da união de conceitos e características dos sistemas conexionistas, sistemas *fuzzy* e sistemas tradicionais de regras de produção,procurando, desta maneira, representar e manipular conhecimento incerto sobre um domínio específico.

Podemos definir o FCM como um grafo direcionado (digrafo) G_d cujos arcos a_{ij} apresentam valores v_{ij} positivos ou negativos do intervalo real [-1,1].

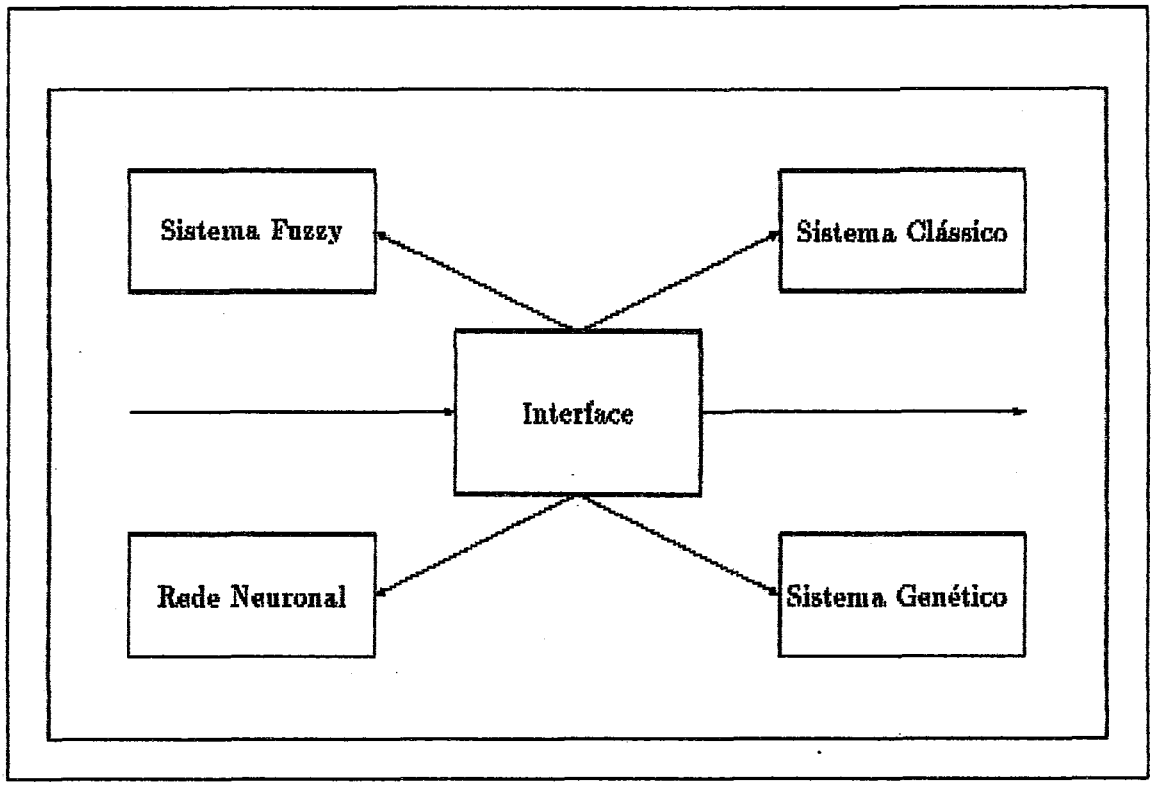

Figura III.1: Esquema dos futuros sistemas híbridos

O arco direcionado a_{ij} de um nó n_i para um nó n_j expressa que existe uma relação causal entre os dois conceitos representados pelos respectivos nós.

O quanto um conceito representado por um nó n_i afeta um outro nó n_j é expresso pelo valor v_{ij} associado ao arco a_{ij} que assume valores no intervalo [-1,1] correspondente ao domínio de um conjunto nebuloso.

Assim, se o valor v_{ij}, associado a um arco a_{ij}, for *maior que zero*, temos que n_i aumenta o efeito expresso por n_j; caso seja *menor que zero*, n_i diminui o efeito expresso por n_j e finalmente, se v_{ij} tiver o valor *igual a zero*, não existe nenhuma relação causal entre n_i e n_j

Vejamos um exemplo bem simples, descrito em [KOSK91], da aplicação FCM para descrever a influência de diversos fatores no equilíbrio político da África do Sul (veja figura III.2).

Os fatores considerados são os seguintes:

C_1: Investimanto Estrangeiro

C_2: Atividade de Mineração

C_3: Nível de Emprego para Negros

C_4: Radicalismo do Racismo pelos Brancos

C_5: Leis de Restrições ao Trabalho Negro

C_6: Unidade das Tribos Negras

C_7: *'Apartheid'*

C_8: Estabilidade do Governo

C_9: Eleitorado do Partido Nacional

- As relações causais positivas são:
  - $C_1 \longrightarrow C_2, C_3, C_8, C_9$
  - $C_2 \longrightarrow C_3, C_8$
  - $C_3 \longrightarrow C_4, C_8, C_9$
  - $C_4 \longrightarrow C_8, C_7$
  - $C_5 \longrightarrow C_6, C_7$
  - $C_6 \longrightarrow C_4$
  - $C_7 \longrightarrow C_5$
  - $C_9 \longrightarrow C_8$
- As relações causais negativas são:
  - $C_3 \longrightarrow C_6$
  - $C_4 \longrightarrow C_9$
  - $C_5 \longrightarrow C_2, C_3$
  - $C_6 \longrightarrow C_7, C_8$
  - $C_7 \longrightarrow C_8$
  - $C_8 \longrightarrow C_7$

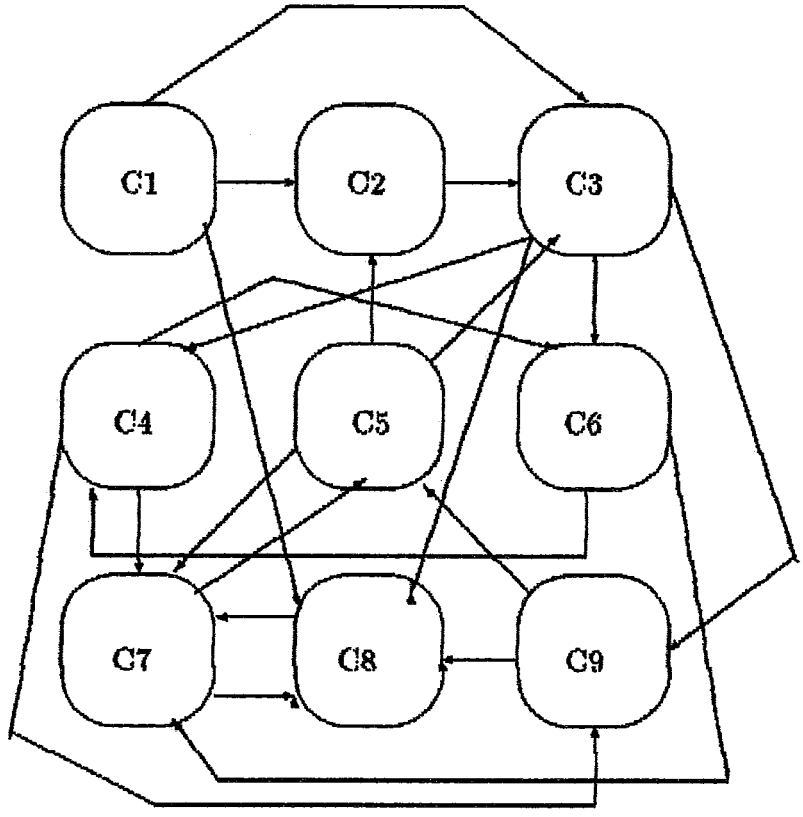

Figura III.2: Esquema das relações causais

– $C_9 \longrightarrow C_5$

Todas estas relações causais podem ser representadas por uma matriz W_{9x9} chamada de *matriz de conexão causal*:

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Considerando a matriz de conexão causal como representante do processo de codificação de um Memória Associativa Temporal - *TAM* [SIMP90] e aplicando o seguinte processo de recuperação de um padrão de atividade:

$$a_j^{k+1}(t+1) = \begin{cases} 1 & \text{se } x_j > 0 \\ a_j^{k+1}(t) & \text{se } x_j \leq 0 \end{cases}$$

onde $x_j = \sum_{j=1}^{n} a_1^k(t) w_{ij}, \forall i, j$ podemos acompanhar a dinâmica do sistema até que este se estabilize.

Por exemplo, se queremos analisar a influência do fator *política de investimento estrangeiro* utilizaremos o vetor :

$$C_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

que representa estímulo a ser aplicado à rede, representada pela matriz de conexão causal.Utilizando-se o processo de recuperação da memória associativa temporal obtemos um novo padrão de atividade que será aplicado novamente até que a rede se estabilize.

No nosso exemplo tivemos o seguinte comportamento da rede:

$$C_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$C_1W = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$C_2 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$C_2W = \begin{pmatrix} 0 & 1 & 2 & 1 & -1 & -1 & -1 & 4 & 1 \end{pmatrix}$$

$$C_3 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$C_3W = \begin{pmatrix} 0 & 1 & 2 & 1 & -1 & 0 & 0 & 4 & 1 \end{pmatrix}$$

$$C_4 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Chegamos, assim, a um estado estável C_3, isto é, a um ponto fixo do sistema dinâmico FCM. Desta maneira, dado o estado inicial C_1, representando a política de investimento estrangeiro, o sistema infere os seguintes estados: $\{C_1, C_2, C_3, C_4, C_8, C_9\}$. Podemos concluir,então, que se a sustentação do investimento estrangeiro for mantida ocorre um estado de equilíbrio entre a estabilidade do governo e o racismo [KOSK91].

O FCM constitui, na realidade, um paradigma para processamento de conhecimento que captura relações causais entre um número arbitrário de variáveis

valoradas por diversos especialistas [SIMP90]. Permite deste modo que várias opiniões de diferentes especialistas sejam integradas em uma única representação, isto é, é possível sintetizar vários FCM's proveniente de vários especialistas em um único FCM.

Outros exemplos de modelos que utilizam a representação puramente local são :

- modelo NETtalk para representar as 7 letras de entrada [SEJN87]
- modelo de percepção de letras em palavras [RUME82]
- modelo TRACE de percepção de fala [MCCL86]
- modelo para processamento de informaçãoes visuais [FELD85]

Resumindo, apesar da simplicidade e facilidade da representação puramente local, além do fato da rede refletir a estrutura do conhecimento representado, este esquema de representação não apresenta muitas propriedades interessantes.

III.3 Representação Parcialmente Distribuída

A noção de representação distribuida é de suma importância para se entender o potencial do enfoque conexionista [SMOL88].Uma analogia interessante foi feita por [PESS89a]. Considerando os dispositivos de memória convencionais, temos que uma informação é armazenada através da cópia desta informação para um determinado lugar. Este lugar é chamado *posição de memória* e é acessado pela especificação do *endereço de memória* desta posição. Desta forma, o processo de acesso a uma posição de memória, isto é, a recuperação de uma informação previamente armazenada ,pode ser visto como *'achando o lugar certo na memória'*. Assim, podemos observar que os sistemas convencionais podem ser caracterizados pelo ser caráter extremamente local. Então, quando utilizamos "frames",por exemplo, teremos que o acesso a um "frame" consiste em selecionar um "frame" específico, que está armazenado em um

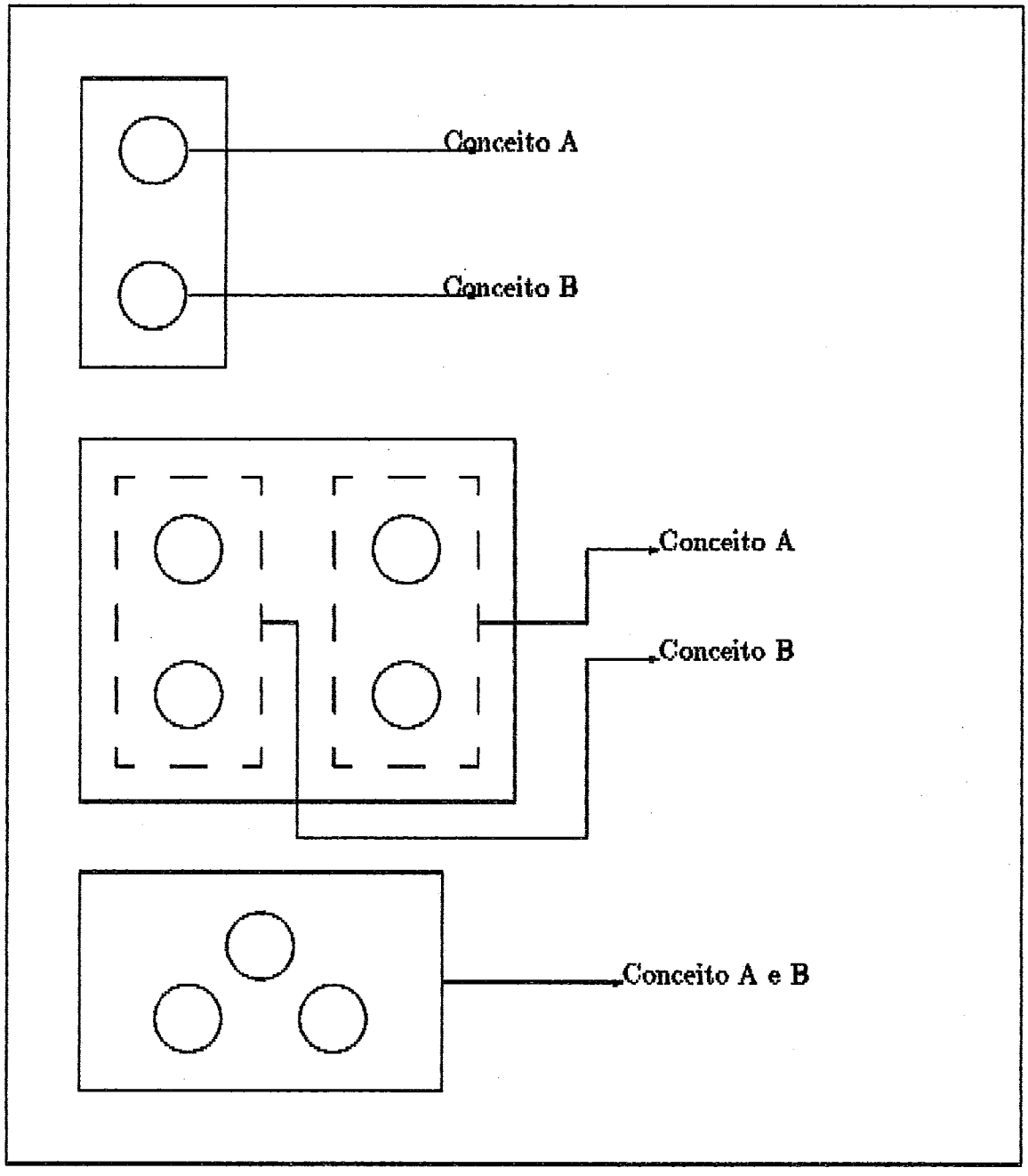

Figura III.3: Tipos de representação: puramente local, parcialmente distribuída e holográfica

lugar endereçavél. O que torna interessante a abordagem conexionista distribuída é o fato da informação não estar associada a um lugar específico, e sim distribuída por diversos lugares. O processo de recuperação que foi visto como *'achando o lugar certo na memória'*, nos sistemas convencionais, passa a ser *'evocando a informação'* nos sistemas conexionistas.

A representação distribuída, assim, é um esquema de memória em que cada entidade (conceito,símbolo, etc) é representado por um padrão de atividade sobre muitas unidades. Se cada unidade participa na representação de muitas entidades, esta unidade é dita *"coarsely turned"* e o esquema de memória constituído por estas unidades é conhecido como *"coarse-coded memory"*

Analisaremos, agora, alguns conceitos relacionados à representação distribuída tomando como base o exemplo dado por [HINT86]. Supondo que desejamos representar uma determinada característica onde é dado um tipo e os valores de alguns parâmetros contínuos que distinguem dois objetos do mesmo tipo.Cada tipo de característica define um espaço de possíveis instâncias e cada parâmetro define uma dimensão do espaço de característica. Por exemplo, se tomarmos pontos de um plano como característica, o espaço de características possíveis é de dimensão igual a 2, já que dois objetos do mesmo tipo (ponto) são diferenciados por dois valores diferentes.Uma maneira de representar este ponto é considerar dois grupos de unidades, *grupo X* e *grupo Y*, que codificarão um certo intervalo dos eixos cartesianos (veja figura III.4). Então, um ponto dado poderia ser codificado pela atividade de 2 unidades, uma do *grupo x* e outra do *grupo y*. Porém, se dois pontos são codificados, seguindo este processo, teremos (x_1, y_1) para o primeiro ponto e (x_2, y_2) para o segundo ponto, o que torna o método inviável, pois na recuperação da informação armazenada teremos quatro possibilidades: (x_1, y_1), (x_2, y_2) e (x_1, y_2) , (x_2, y_1) (veja figura III.5). Desta forma, não possuimos meios de saber quais os pontos realmente armazenados. Este problema é conhecido como o problema da ligação *('binding problem')* que é de fundamental inportañcia para a manipulação de estruturas simbólicas por sistemas conexionistas [SMOL87],[SMOL87b],[SMOL88]

Um conceito importante à analise de esquemas de representação distribuída é o conceito de precisão. Definimos a precisão como sendo o número de

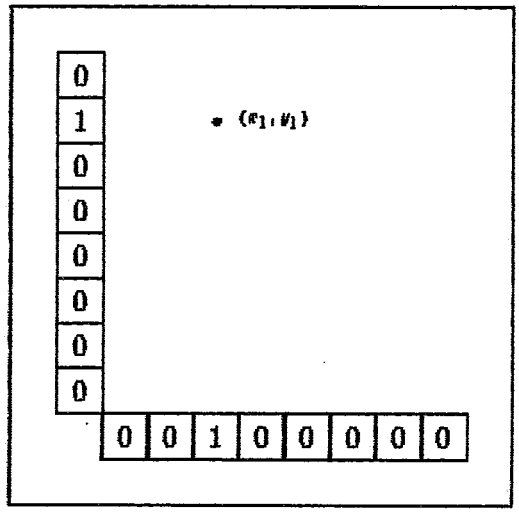

Figura III.4: Forma simples de codificar um ponto no espaço bidimensional utilizando-se dois grupos de unidades

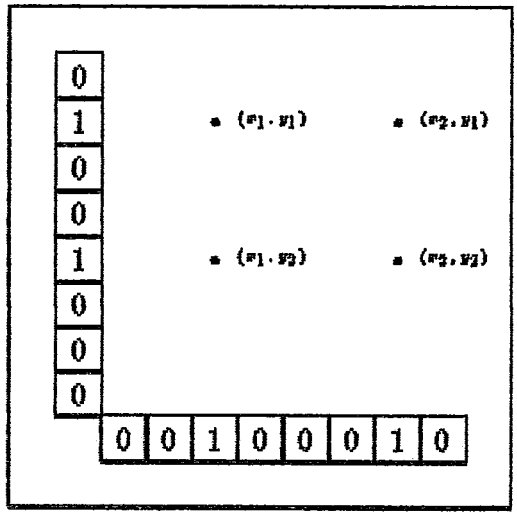

Figura III.5: As quatros possibilidades para a representação de dois pontos

diferentes codificações que são geradas quando um ponto é deslocado uma distância padrão através do espaço.Considerando esta definição e o esquema exposto, temos que a precisão é proporcional ao número de unidades utilizadas na representação.

III.3.1 'Coarse Coding'

A técnica conhecida como *'coarse coding'* foi desenvolvida por Hinton com objetivo de reduzir o número de unidades necessárias para representar uma faixa de valores com uma precisão pré-determinada [HINT80]. Vários sistemas conexionistas para processamento simbólico utilizam esta técnica de representação ([TOUR88],

[TOUR90], por exemplo). Ela pode ser analisada da seguinte maneira: considera-se o espaço bidimensional (plano) estando dividido em regiões circulares (de mesmo raio) distribuídas randomicamente. Para cada região circular é associada uma unidade. Quando movemos um ponto através do espaço uma distância padrão, geramos um conjunto de códigos que descrevem cada ponto da trajetória. Como observado anteriormente o número de elementos pertencentes a este conjunto de códigos gerados é o que chamamos de precisão. Analisaremos, agora, o deslocamento de um ponto em linha reta através do espaço (bidimensional, no caso). Temos que cada vez que o ponto atravessa um limite da região circular há uma mudança no estado de atividade da unidade correspondente, gerando assim um nova codificação. Desta maneira o número de mudanças observadas é igual ao dobro do número de regiões que o ponto atravessa. Além disso, todas as regiões circulares que apresentam uma distância do centro ao ponto menor que o raio, participarão da codificação deste ponto. Assim, concluimos que a precisão p é proporcional ao número de regiões n_r e ao raio r destas regiões,isto é, a precisão deste esquema de representação é proporcional ao número de unidades e o raio das regiões circulares.

Em [ROSE88] temos uma análise matemática das relações entre diversos parâmetros que constituem o modelo geral de um tipo de *"coarse coding"* denominado CCSM (*"Coarse-Coded Symbol Memories"*).

Até então, vários parâmetros (capacidade da mémoria, por exemplo) eram obtidos por testes empíricos e o comportamento desejado era conseguido por tentativa e erro.

Dentro do âmbito das memórias *"coarse-coded"* encontramos diversos paradigmas, dentre os mais representativos temos:

Representação Baseada em Características (RBC): Neste paradigma temos que cada unidade participa da representação de uma característica semântica do objeto que está sendo representado. Enquanto que na representação local temos uma unidade para cada objeto, neste tipo de representação temos que várias unidades representam características semânticas de vários objetos e cada objeto representado é definido por um conjunto de características

semânticas, isto é , como a interseção de diversas características/unidades que constituem este objeto. Unidades binárias podem, assim, codificar características semânticas binárias, ao passo que características mais complicadas (propriedades multivaloradas, por exemplo) requerem unidades mais elaboradas ou grupos de unidades. No caso das unidades binárias temos que a similaridade entre os conceitos compostos por características binárias podem ser mensuradas pela distância de Hamming entre suas representações [ROSE88]. Uma propriedade interessante deste tipo de representação e a emergência da propriedade de generalização.

"Coarse-Coded Symbol Memories"- (CCSM): Neste paradigma cada símbolo (objeto , conceito) é representado por um conjunto arbitrário de unidades formando o que chamamos de *padrão*. Ao contrário do paradigma anterior, as unidades não possuem nenhuma relação semântica com o símbolo representado. Um símbolo é armazenado na memória pela ativação de todas as unidades que formam o seu padrão. Consequentemente, para que um símbolo seja considerado presente na memória é necessário que todas as unidades associadas ao seu padrão estejam ativadas. O *campo receptivo* de uma unidade é definido como o conjunto de todos os símbolos em que esta unidade participa na formação do padrão associado ao símbolo. Uma das características deste tipo de representação é a eficiência com que ela manipula memórias esparsas o que justifica sua utilização em diversos modelos [ROSE85]

Embora as representações *"coarse-coded"* apresente uma inovação quanto às formas mais antigas de representação ainda subsistem diversos problemas [POLL90]:

1. A representação *"coarse-coded"* necessita de mecanismos complexos de acesso tais como *"pullout networks"* e espaço de cláusulas (trataremos disso detalhadamente no Capítulo V).

2. CCSM só podem instanciar um pequeno número de elementos representacionais antes que elementos espúrios apareçam (efeito fantasma).

3. Necessitam de um número extremo de unidades (centenas ou milhares)

III.3.2 A Questão das Micro-características

Conforme ressaltado por [SHAR91] o termo *micro-característica* não tem sido utilizado de forma totalmente consistente na literatura.

Existe o conceito de micro-característica aceito pela maioria dos pesquisadores que as vê como elementos atômicos da representação distribuída. A divergência surge quanto à interpretação semântica das micro-características. Alguns autores ([MCCL86], por exemplo) utilizam o conceito de micro-característica que por si só são semanticamente interpretado sem a consideração do papel de cada micro-característica no conjunto total da representação, isto é, é possível extrair informações da rede somente observando a representação, não necessitando que se treine a rede para interpretá-la. Outros autores ([HINT81], por exemplo) usam o termo micro-característica para se referirem a elementos individuais que não são semanticamente interpretados sem a participação do processamento de toda a representação. Isto significa que no primeiro caso temos as micro-características como propriedades que estão presentes no mundo (*é humano, é mortal*,etc) e são conhecidas como micro-características simbólicas. No segundo caso, temos que as micro-características não correspondem às propriedades que estão presentes no mundo. Estas micro-características são classificadas como não simbólicas e são interpretadas como propriedades emergentes do padrão de ativação do conjunto de micro-características como um todo.

III.3.3 Esquemas Conexionista

Um exemplo da utilização da representação parcialmente distribuída em modelos conexionistas, e no caso, da representação baseada em características (RBC) é o estudo dos esquemas conexionistas.

Um "*esquema é uma descrição simbólica e estereotipada das características de objetos ou conceitos. Esquemas expressam, através de um conjunto de*

descritores específicos, o que há de típico nas situações representadas " [PESS90b]. Esta definição nos precisa o caracter de alto nível de abstração desta forma de representação. Apesar do sucesso em certas aplicações específicas e de domínio restrito a classe de esquemas, onde os *"frames"* de Minsky [MINS75], *"scripts"* de Schank [SCHA77] e os *"schemata"* de Bobrow e Norman [BOBR75] estão incluso, não se mostrou adequada para capturar e manipular a incompletude e a incerteza inerente ao conhecimento humano.

A abordagem conexionista dos esquemas tenta construir uma solução em que se utilizem os conceitos e propriedades dos modelos simbolistas tradicionais,tais como [PESS89b]:

- esquemas possuem variáveis
- esquemas são formados por sub-esquemas
- esquemas podem representar vários níveis de abstração
- esquemas são dispositivos de reconhecimento

Assim, os esquemas conexionistas tentam combinar as caracteristicas da representação tradicional com a flexibilidade, capacidade de aprendizado e adaptação, etc das redes neuronais.

Para uma visão mais concreta desta problemática construiremos um modelo de um esquema conexionista que represente um pequeno domínio de conhecimento objetivando ressaltar as características, vantagens e desvantagens desta abordagem representacional.

O domínio da representação

Utilizaremos um conjunto de instrumentos musicais composto dos seguintes objetos:

- violino
- violão

- violoncelo
- flauta
- gaita
- tuba
- prato
- tímpano

Os objetos são descritos pelos seguintes conjunto de descritores:

1. instrumento musical
2. instrumento de corda
3. instrumento friccionado
4. instrumento de porte pequeno
5. instrumento feito de madeira
6. instrumento dedilhado
7. instrumento de porte médio
8. instrumento de porte grande
9. instrumento de embocadura livre
10. instrumento de sopro
11. instrumento de palheta dupla
12. instrumento de metal
13. instrumento feito de couro
14. instrumento de bocal
15. instrumento de percussão

instrumento	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
violino	+	+	+	+	+	-	-	-	-	-	-	-	-	-	-
violão	+	+	-	-	+	+	+	-	-	-	-	-	-	-	-
violoncelo	+	+	+	-	+	-	-	+	-	-	-	-	-	-	-
flauta	+	-	-	+	-	-	-	-	+	+	-	-	-	-	-
gaita	+	-	-	-	+	-	+	-	-	+	-	-	+	-	-
tuba	+	-	-	-	-	-	-	+	-	+	-	+	-	+	-
prato	+	-	-	-	-	-	+	-	-	-	-	+	-	-	+
tímpano	+	-	-	-	-	-	-	+	-	-	-	+	-	-	+

Tabela III.1: Tabela de padrões a serem aprendidos

Se tivermos o seguinte padrão " + + + + + - - - - - - - - - - " presente na saída da rede, temos então a representação de um instrumento musical de corda,friccionado, de porte pequeno e feito de madeira. No nosso domínio esta descrição corresponde ao violino.

O modelo da rede neuronal

O modelo escolhido para a simulação foi o *"Brain-State-in-the-Box"* (BSB) desenvolvido por James Anderson e colaboradores [ANDER77]. O modelo BSB é essencialmente uma rede associativa linear combinada com um algoritmo não linear de pós-processamento responsável por limpar respostas espúrias ,não desejadas, produzida pela rede linear. Na realidade, o que temos é um sistema linear com realimentação positiva e potencialmente instável e com a introdução de uma simples não linearidade (limite na *"firing rate"* de cada unidade) o sistema fica limitado a um hipercubo (daí o termo *brain-state-in-the-box*) e a convergência é garantida pelo teorema de Cohen-Grossberg [COHE83].

O modelo *"Brain-State-in-the-Box"* e o modelo Hopfield [HOPF82] são bem similares, já que pertencem a uma classe mais geral denominada modelos de satisfação de restrições, entretanto, o modelo BSB não exige que a rede seja completamente interconectada e as interconexões existentes não precisam ser simétricas, isto é, a matriz de pesos não necessita ser simétrica (veja figura III.6). Esta carac-

terista é interessante quando desejamos dar interpretaçãoes semânticas aos elementos da rede [PESS90b].

O algoritmo BSB

O BSB é uma rede associativa linear (mais especificamente, autoassociativa) que é expressa por uma multiplicação matricial, isto é, se x é um vetor que representa um estímulo de entrada e W é a matriz de conexão sináptica então y , o vetor resposta ao estímulo x, é:

$$y = Wx$$

O caracter linear dos modelos associativo, e mais precisamente o princípio da superposição, permite a interpretação do modelo como mecanismo simples de um raciocínio cooperativo. Assim, se temos que:

$$Wx_1 = y_1 + y_2 \text{ e } Wx_2 = y_2 + y_3$$

então,

$$W(x_1 + x_2) = y_1 + 2y_2 + y_3$$

o que nos mostra que se apresentado dois estímulos de entrada juntos teremos que a resposta produzida reforça a associação comum a ambos ($2y_2$). O que o pós-processamento do algoritmo BSB faz é retirar as respostas que não foram reforçadas (no caso, y_1 e y_3).

O procedimento de aprendizado do BSB utiliza a regra de correção de erros:

$$\Delta w_{ij} = \alpha u_i^k d_j,$$

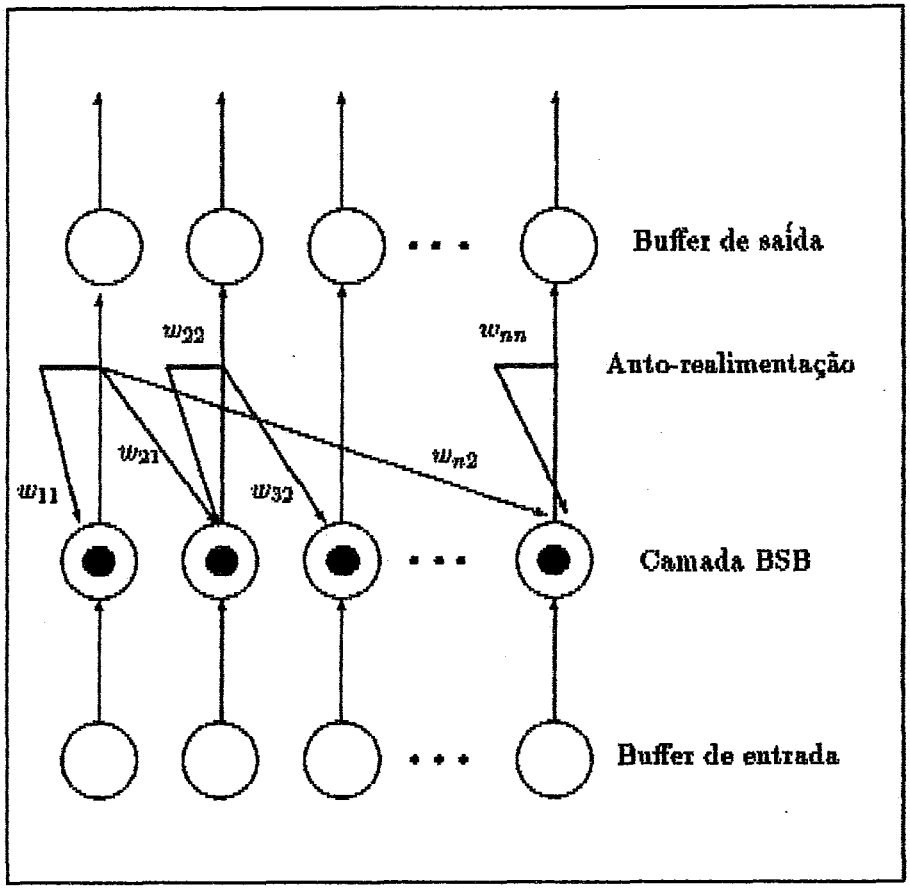

Figura III.6: Esquema da rede *"Brain-State-in-the-Box"*

onde Δw_{ij} é a variação no peso entre a unidade u_i e u_j, α é uma constante positiva que controla a taxa de aprendizado, e d_j é expresso pela equação:

$$d_j = u_j^k - \sum_{i=1}^{n} w_{ij} u_i.$$

O processo de aprendizado se processa pela aplicação da regra de correção de erros para cada padrão até que o o erro d_j produzido por cada saída seja suficientemente baixo [SIMP90].

Para a recuperação de uma informação armazenada o BSB utiliza uma função limiar rampa em uma operação de realimentação. Ao se apresentar um padrão de entrada o BSB utiliza a seguinte equação:

$$u_i(t+1) = f(u_i(t) + \beta \sum_{i=1}^{n} w_{ij} u_j(t)),$$

onde a função limiar rampa é:

$$u_j(t+1) = \begin{cases} +\gamma & \text{se } x_j \geq \gamma \\ u & \text{se } \mid x \mid < \gamma \\ -\gamma & \text{se } x_j \leq \gamma \end{cases}$$

β é a constante que controla a realimentação e γ os limites da imagem de f.

A simulação

A simulação teve como objetivo a observação de propriedades significativas da representação distribuída e não o estudo da performance de algoritmos ou novas técnicas de representação.

Utilizamos o simulador NeuralWare versão 2.0 que é de fácil utilização permitindo uma ampla experimentação através da alteração de vários parâmetros em grupos ou isoladamente. O simulador fornece uma documentação automática da rede construída permitindo a reprodução da mesma em outros simuladores. Os parâmetros e o conjunto de dados para o aprendizado da rede se encontram no Apêndice 2.

Vejamos agora o comportamento da rede. A fim de se explorar as características dos esquemas conexionista realizamos alguns testes. Um dos teste foi a determinação de valores default para esquemas que apresentavam mais de um valor possível. Por exemplo, a rede aprendeu que os instrumentos musicais de corda, feitos de madeira pode ser: violino, violão e violoncelo. Qual é o valor default para este caso? A determinação de valores default é que constituem este teste, e neste caso obtivemos que o valor default para o instrumento de corda feito de madeira é o violão, isto é, quando ativamos a unidade 1 (instrumento musical) e a unidade 5 (instrumento feito de madeira) e consideramos as outras com valor 0. Assim, após o processamento da rede, dada a entrada anterior, obtivemos o seguinte padrão "+ + - - + + + - - - - - - - -", em outras palavras, o padrão representativo do violão. Outros casos verificados podem ser encontrados nas tabelas III.2 e III.3. O outro teste foi a determinação do instrumento default para a rede, que no caso é a gaita.

Com esse exemplo simples explicitamos uma das formas da utilização da representação parcialmente distribuída em modelos conexionista onde as micro-

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
entrada exp1	+	+	.	.	+	.	.	.	.	.	.	.	.	.	.
saída exp1	+	+	-	-	+	+	+	-	-	-	-	-	-	-	-
entrada exp2	+	.	.	.	.	.	.	.	.	+	.	.	.	.	.
saída exp2	+	-	-	-	-	-	-	+	-	+	-	+	-	+	-
entrada exp3	+	.	.	.	.	.	.	.	.	.	.	+	.	.	.
saída exp3	+	-	-	-	-	-	+	-	-	-	-	+	-	-	+
entrada exp4	+	.	.	.	.	.	.	.	.	.	.	.	.	.	.
saída exp4	+	-	-	-	+	-	+	-	-	+	-	-	+	-	-

Tabela III.2: Tabela de experimentos para verificação de valores default

instrumento de corda/madeira default	violão
instrumento de sopro default	tuba
instrumento de percussão default	prato
instrumento default	gaita

Tabela III.3: Tabela de valores default

características, que descrevem o domínio, podem ser interpretadas semanticamente. Um outro ponto que deve ser destacado é a capacidade de generalização deste tipo de representação, isto é, a capacidade de generalizar certos conceitos que no nosso caso são os valores default. Para um estudo mais detalhado dos esquemas conexionista veja [PESS89b] e [PESS90b].

III.4 Representação Holográfica

A representação holográfica ou totalmente distribuída estabelece que cada conceito será representado por todas as unidades da rede. Uma das fundamentações teóricas para esta abordagem,isto é, estudos que indicam a plausabilidade biológica desta maneira de modelar o cérebro, pode ser encontrada no clássico *"Languages of the Brain"* do neurocientista Karl Pribram [PRIB71].

O um dos conceitos mais importante na abordagem holográfica e a de *holograma*. *"Holografia é um método de fotografia sem lente no qual o campo*

ondulatório da luz espalhada por um objeto é registrado numa chapa sob a forma de um padrão de interferência. Quando o resgistro fotográfico - o holograma - é exposto a um feixe de luz coerente, como o laser, o padrão ondulatório original é regenerado. Uma imagem tridimensional aparece. Como não há focalizador, isto é, lentes focalizadoras, a chapa tem a aparência de um padrão de espirais destituído de qualquer significado. Qualquer pedaço do holograma pode reconstruir a imagem inteira" [WILB82].

A proposta holográfica tenta explicar como as informações se distribuem pelo cérebro, a localização específica de certas funções, a enorme capacidade de armazenamento de informações em um volume limitado,etc. A hipótese holográfica não pretende ser a grande solução teórica para a explicação total da fisiologia cerebral e dos processos cognitivos mas apresenta convincentes dados experimentais e descrições matemáticas para sistemas visuais, auditivos e somatossensoriais além oferecer novas pespectivas de pesquisa.

Embora a teoria holográfica tenha sido recebida com uma certa reserva, pois era muita ampla, permitia extrapolações metafísicas e não possuia uma tecnologia associada, ela ressurge através da união com as redes neuronais e a tecnologia da computação ótica [WASS89],[KIM90], [WEST91].

Assim, a abordagem holográfica/conexionista ainda se encontra em gestação e diversos problemas associados a esta forma de representação ainda se apresentam insolúveis: a representação de diversos conceitos de maneira não sequencial, o controle da interferência devido ao alto grau de distribuição, etc.Os trabalhos ainda gravitam em torno dos mecanismos de mémoria associativa holográfica [WILL81], [PAO89], [HINT81], sendo inexistente abordagens mais complexas a nível de representação de conhecimento, sendo a única excessão os trabalhos desenvolvido por Paul Smolensky, que estudaremos no capítulo VI

Capítulo IV

Processamento Simbólico Conexionista

IV.1 Modelos Conexionistas de Alto Nível

Vários são os problemas de difícil solução e abordagem enfrentados pela ótica conexionista. Representação do conhecimento estruturado, inferência, gerência de instanciação de estruturas simbólicas são alguns exemplos desses problemas [POLL88]. Os modelos que tratam tais problemas são conhecidos por *modelos conexionistas de alto nível.*

Uma justificativa para o desenvolvimento de modelos conexionistas de alto nível é a tentativa de reunir dois níveis semânticos em um só modelo . Modelos conexionistas distribuídos podem apresentar representações que possuam uma *micro-semântica*, isto é, símbolos com representações internas similares tendem a ter efeitos de processamento similares. Já os símbolos nos sistemas tradicionais não possuem esta *micro-semântica* ,haja visto que são formados pela concatenação de códigos ,por exemplo ASCII, que são estáticos. Estes símbolos tradicionais possuem uma *macro-semântica*, isto é, um conjunto arbitrário de bits (constituindo um símbolo) é armazenado em estruturas relacionadas com outros símbolos, através de ponteiros, e ligações são propagadas através de variáveis [DYER90]

Uma forma comumemte utilizada para a investigação do processamento simbólico conexionista é através da pesquisa em processamento da linguagem

natural,e no caso, estes estudos são designados como processamento conexionista em linguagem natural (PCLN),isto é, estão no âmbito da pesquisa do PLN com métodos conexionistas. Este trabalho não tem o objetivo de abordar questões desta linha de investigação extremamente rica e complexa e sim de se utilizar de exemplos do PCLN que realce pontos significativos das técnicas de representação do conhecimento.

O PLN requer técnicas que trabalhem com estruturas, regras, etc que são sustentadas pelos sistemas tradicionais de processamento simbólico. Apesar das facilidades fornecidas pelas técnicas tradicionais os sistemas de PLN tendem a ser fragéis, restrito a domínios específicos e sofrerem do gargalo da engenharia do conhecimento, isto é, grande parte do conhecimento é construído manualmente.

Os sistemas conexionistas apresentam características que se combinadas adequadamente com abordagens clássicas talvez possam abrir caminho para a solução de diversos problemas inerente ao processamento da linguagem natural. Mesmo que não se resolva completamente tais problemas é possível que se atenue uma série de características indesejáveis dos sistemas estritamente simbólicos.

Além disto, o processamento conexionista da linguagem natural é um dos campos de pesquisa que mais tem contribuído para o desenvolvimento de técnicas de representação que não possuem paralelo com as representações tradicionais [SHAR91].

IV.1.1 Características dos Modelos Conexionistas e Simbolistas

Cada enfoque (conexionista ou simbolista) apresenta um conjunto de características fundamentais para um sistema híbrido. Vejamos algumas destas caracteristicas:

- Características dos sistemas conexionistas

  aprendizado automático e generalização: comportamento não algoritmico e sim resultado do aprendizado; capacidade de generalizar estruturas estatísticas a partir do conjunto de dados utilizados durante o processo de treinamento da rede.

memoria associativa, paralelismo e tolerância a falhas: memória que permite reconstruir, a partir de fragmentos, o padrão original; degradação suave quanto à falhas de hardware.

plausabilidade neuronal: inspiração biológica que pode ser refinada em trabalhos futuros.

- Características dos sistemas tradicionais

"tokens" e *"types"*: é possível a construção dinâmica de instâncias que se mantem distintas de um tipo geral.

herança: dados *"types"* e *"tokens"*, é possível uma atualização de um determinado *"type"* que é imediatamente dedutível para todos *"tokens"* relevantes que são instâncias deste *"type"*.

referência virtual: uma estrutura simbólica pode apontar para outra estrutura que está em um local distante da memória física, permitindo, assim, a criação de complexas memórias virtuais sem a necessidade de uma reorganização da memória física.

estutura e compositabilidade: utilizando-se ponteiros é possível construir estruturas recursivas; é possível, também, através de fatos e regras a produção de um número potencialmente infinito de novas estruturas ,por composição repetida, a partir de uma estrutura inicial.

variáveis e operações sensíveis a estrutura: através de variáveis e estruturas pode-se especificar um número infinito de instâncias.

comunicação e controle: Nas linguagens que manipulam estruturas (LISP, por exemplo) é possível a comunicação a nível estrutural.

gerenciamento de memória: em sistemas simbolistas existe sempre funções que fornecem memória conforme as necessidades, em tempo de execução, para a construção de novas estruturas, dentro de certos limites, é claro.

IV.2 Recirculação Simbólica

O fato de que sistemas conexionistas e sistemas tradicionais de processamento simbólico possuirem características distintas e extremamentes úteis em tarefas cognitivas de alto nível tem motivado uma forte pesquisa a procura de uma síntese. O que é considerado necessário para esta síntese é um método em que símbolos criam dinamicamente sua própria micro-semantica enquanto ao mesmo tempo formam relações recursivas e estruturas com outros símbolos, assim, desenvolvendo também uma macro-semântica.

Uma técnica que tenta realizar esta síntese é a técnica conexionista da Recirculação Simbólica (*"Symbol Recirculation"*). Este método utiliza uma rede neuronal designada por *"global symbol lexicon"* responsável pelo armazenamento e manutenção das representações dos símbolos. Cada símbolo é composto por um padrão de ativação distribuído pelas unidades da rede. Inicialmente as representações dos símbolos são padrões de ativação randômicos. As representações dos símbolos surgem como resultado do treinamento da rede para formar mapeamentos associativos com os outros símbolos, sendo que estes mapeamentos são responsáveis também pela captura das relações estruturais.

A técnica básica da recirculação simbólica pode ser descrita como constituída das seguintes etapas:

1) Inicialização da rede *"global symbol lexicon"* com uma representação arbritária para cada símbolo.

2) Carregamento destes símbolos para camadas de entrada/saída de uma ou mais redes neuronais.

3) Modificação das representações dos símbolos na rede *"global symbol lexicon"* a fim de auxiliar a(s) rede(s) de mapeamento no desempenho da tarefa, enquanto, ao mesmo tempo, modificam-se os pesos das conexões desta(s) rede(s).

4) Armazenamento das representações modificadas dos símbolos de volta na rede *"global symbol lexicon"* como padrões de atividades.

5) Iteração sobre todos os símbolos, redes e tarefas até que todas as representações simbólicas tenham estabilizado para todas as tarefas.

Um fato que deve ser observado é que um mesmo vetor de valores pode ser visto, a qualquer momento, de duas maneiras:

1) **Conjunto de pesos:** pesos, estes, que serão ajustados durante o aprendizado.

2) **Um padrão de ativação:** que será apresentado como entrada/saída para alguma rede.

Resumindo, através da recirculação simbólica obtemos um processo dinâmico onde as representações são modificadas durante o processo de aprendizado, enquanto nos métodos conexionistas tradicionais o conjunto de treinamento se mantém inalterado durante todo o processo de aprendizado. Um exemplo da utilização desta idéia (recirculação simbólica) pode ser encontrada em [MIIK91] e a descrição do procedimento de aprendizado pode ser encontrado em [HINT88].

IV.3 A Questão da Modularidade

Os sistemas conexionistas tradicionais quase sempre não apresentam uma arquiterura modular e quando apresentam estas são extremamente simples. Segundo [MIIK91] esta abordagem não-modular dos sistemas conexionistas não consegue grandes avanços na solução de problemas cognitivos de alto nível, estando assim, limiatada a problemas ainda considerados de baixo nível.

Ele aponta três razões principais que são responsáveis pelo fraco desempenho das redes homogêneas, isto é, não modulares, na modelagem das tarefas de alto nível:

1. Tarefas de alto nível usualmente requerem a composição de sub-tarefas, isto é, um composição de vários sub-processos distintos e específicos que devem trabalhar de forma coordenada e cooperativa na solução de um determinado problema.

2. Conforme o aumento do tamanho da rede (acompanhando a complexidade do problema, normalmente) o número de exemplos para o aprendizado e o tempo necessário para tal aprendizado aumentam de forma que se tornam intratáveis, especialmente nos problemas de processamento simbólico sequencial conexionista.

3. Não existe um método específico que permita um acompanhamento do que e como o sistema, num todo, está realizando, isto é, qual o conhecimento que está sendo adquirido e aplicado durante a solução de um determinado problema .

Resumindo, os sistemas conexionistas tradicionais não apresentam uma arquitetura modular que permita a realização de tarefas que por si só encaminham uma solução modular e estruturada.A observação de como (e qual) o conhecimento é adquirido e utilizado apresenta uma complexidade difícil de ser analisada e controlada. Dyer sugere que a construção de sistemas conexionistas modulares,isto é, sub-redes neuronais que trabalham em conjunto para realizarem uma tarefa de alto nível, contornará alguns obstáculos que a abordagem tradicional se mostra inadequada a tratar.

A abordagem modular conexionista começa a ser explorada e novas questões se apresentam, tais como:

- Como uma tarefa deve ser decomposta em módulos e que organização deve existir entre os módulos ?
- Como essas redes modulares devem ser projetadas para servirem de blocos básicos reutilizáveis por outros sistemas ?
- Como se dá a comunicação e o controle entre os módulos ?
- Como funciona o aprendizado nos sistemas modulares ?

Estas questões, ainda em grande parte, não apresentam soluções bem estabelecidas e alguns trabalhos apresentam soluções iniciais para esta nova abordagem conexionista [SMIE92].

IV.4 Algumas Propostas para o Processamento Simbólico Conexionista

Podemos considerar 5 principais propostas ou método para a exploração do processamento simbólico conexionista:

- Sistema de produção conexionista distribuído (*"Distribued Connectionist Production System"*)
- Gramática de Grafos [CARD90]
- Representação de hierarquias via descrições reduzidas (*"Representing Hierarchies via Reduces Descriptions"*)
- Memória auto-associativa recursiva (*"Recursive Auto-Associative Memory"*)
- Representação por produto tensorial (*"Tensor Product Representation"*)

O sistema de produção conexionista distribuído desenvolvido por Touretzky e outros será estudado no capítulo V e a representação por produto tensorial, que é o foco principal desse trabalho, será discutida no capítulo VI.

IV.4.1 Representação de hierarquias via descrições reduzidas

A representação de hierarquias via descrições reduzidas está calcada na hipótese que padrões de atividade de algumas partes de uma rede neuronal precisam possuir duas características de um símbolo: deve permitir o acesso a uma representação completa de que faz parte; e ser um descrição reduzida de um objeto.

Hinton explica esta idéia pela analogia com a implementação de estruturas de dados hierárquicas nos sistemas tradicionais. Uma estrutura (*struct* na linguagem C e *record* na linguagem Pascal) é formada por um conjunto prédeterminado de campos e ponteiros para outras instâncias desta estrutura ou um objeto primitivo. Desta maneira é possível a construção, de forma flexível, de estruturas de dados hierárquicas pela dinâmica característica dos ponteiros.Assim, temos

que endereços atuam como símbolos para expressões, e segundo Hinton, ilustram a essência de um símbolo: "

*'e uma representação reduzida de um objeto que provê um acesso remoto a representação completa deste mesmo objeto".*De uma forma geral, esta representação completa é por si só compostas de representações reduzidas, isto é, endereços de estruturas que preenchem os campos do registro. Devido esta representação reduzida é possível a representação de estrutura mais complexa pela utilização da representação completamente articulada, isto é, representação reduzida da estrutura por meio de ponteiros.

Normalmente, quando utilizamos endereços como símbolos não possuímos nenhuma informação sobre o que esses endereços (símbolos) representam. Mas existem casos em que isto não é verdade. Por exemplo, podemos ter um tipo de estrutura de dados que é armazenado na metade superior de uma memória e outro tipo que é armazenado na parte inferior desta mesma memória.Ora, ao examinarmos os bits do endereço de um símbolo podemos ter um descrição do tipo de dado a que este símbolo se refere. Desta maneira, é possível a verificação de um tipo sem a verificação do conteúdo referenciado pelo ponteiro. Assim, este símbolo pode ser visto como uma descrição reduzida de um objeto. Para um estudo mais aprofundado da técnica proposta por Hinton veja [HINT90].

IV.4.2 Memória Auto-Associativa Recursiva - *"RAAM"*

A memória auto-associativa recursiva foi desenvolvida por Pollack [POLL90] com o objetivo de explicar como a descrição reduzida de Hinton [HINT90] pode ser aprendida, já que tal questão não fora desenvolvida em detalhes anteriormente [SHAR91]. O método desenvolvido visa treinar uma rede neuronal que funciona como uma pilha (*"stack"*) e que seja capaz de realizar as operações básicas sobre esta estrutura de dados não-primitiva, ou seja, as operações de colocar um elemento na pilha (empilhar) e retirar um elemento da pilha (desempilhar).

Pollack afirma que a arquitetura *"RAAM"* possui as seguintes propriedades:

- As codificações são desenvolvidadas mecanicamente por uma rede adaptativa.
- Os mecanismos de acesso são simples e determinísticos.
- Um grande número de elementos primitivos podem ser seletivamente combinados em constituintes estruturais.
- A representação utiliza um número extremamente pequeno de unidades.
- O modo de agregação é composicional.

O problema tratado por Pollack pode também ser visto como a tentativa de representar seqüências de símbolos ou árvores de tamanho variável em uma rede neuronal de tamanho fixo ([GELD90], [SHAR91]).

Na arquitetura *"RAAM"* o estado da pilha, em um determinado instante, corresponde a um padrão particular sob um conjunto de 10 unidades da camada escondida. Colocar um novo elemento na pilha (operação de empilhar) significa a gerar um novo padrão correspondendo à expansão da pilha. Este novo padrão é formado pela rede em um processo que combina o estado anterior da pilha com o novo elemento. O processo de retirada da pilha é simplesmente o processo inverso. A partir do padrão da pilha a rede gera dois novos padrões: o padrão correspondente ao elemento do topo da pilha e o padrão correspondente da pilha menos o elemento do topo. Para a construção de árvores se utilizam várias pilhas que podem ser construídas recursivamente

Capítulo V

Sistema de Produção Conexionista Distribuido

V.1 Introdução

Neste capítulo,inicialmente, apresentaremos os principais conceitos referentes a um sistema de produção clássico e a seguir descreveremos e discutiremos o *DCPS* ("Distribued Connectionist Production System"). O objetivo da apresentação deste modelo è o fato de que precisamos de um referencial para diversas análises e comparações , além do *DCPS* já ser considerado uma abordagem clássica para a representação estruturada em modelos conexionistas. Consequentemente, diversos trabalhos nesta linha temática partem dos resultados e limitações deste modelo.

Trataremos dos pontos principais da arquitetura do *DCPS* para que possamos compreender a dinâmica do funcionamento do sistema que será trabalhado no próximo capítulo.

V.2 Sistema de Produção Clássico

Sistema de Produção (SP) é um conceito simples que tem sido bastante utilizado recentemente em larga faixa de aplicações, abrangendo desde investigações sobre a inteligência humana até construção de sistemas especialistas [NILS80],[BUCH84]. Um fato que contribui para esse panorama é a maior facilidade de capturar e codificar o conhecimento quando comparado com outros modelos de computação.

Desde a primeira proposta dos SP's como mecanismo geral de computação feita por [Post 43], estes têm sofrido um grande desenvolvimento. Em 1960, Newell e Simon utilizaram a generalidade das representações *"if-then"* em uma estrutura de controle dirigida por dados. A partir daí, desenvolvendo-se o presente conceito de um sistema de produções que vê a computação como um processo de manipulação de regras em uma ordem determinada pelo dado. Este enfoque é fundamentalmente diferente da forma de uma seqüência de controle encontrada nos programas convencionais.

Assim, o que temos hoje é basicamente uma versão operacional do formalismo lógico de um sistema baseado em regras *"if-then"* que indica como cadeias de símbolos podem ser convertidas em outros símbolos [COST86]. Este formalismo proposto por Post equivale a todas as caracterizações formais para computabilidade tais como : a máquina de Turing, cálculo lambda, recursivas parciais. algoritmos markovianos etc.

V.2.1 Estrutura de um Sistema de Produção

Apesar das variações apresentadas por diversas implementações, a estrutura de um SP é basicamente uma só:

- Conjunto de regras (ou produções) que constituem a memória de produção - *PM ("Production Memory").*
- Base de dados global ou memória de trabalho - *WM ("Working Memory").*
- Interpretador ou máquina de inferência - *IE ("Inference Engine").*

Produções

Um produção é um operador do tipo :
if $C_1 \& C_2 \& \ldots \& C_n$ then $A_1; A_2; \ldots; A_m$

onde C_1 até C_n são condições e A_1 até A_m são ações ou condições.

O lado esquerdo - *LHS ("Left Hand Side")* é também chamado de antecedente e o lado direito - *RHS ("Right Hand Side")* de consequente. Uma produção deve ser interpretada da seguinte maneira: se todas as condições da lista de condições ($C_1, \ldots C_n$) forem verdadeiras no estado atual da base de dados, então executar todas as ações da lista de ações ($A_1, \ldots, A_m$).

Memória de Trabalho

A memória de trabalho é simplesmente a coleção de símbolos - *WME ("Working Memory Elements")* que tenta refletir o estado atual do mundo (domínio de aplicação). Consequentemente, a interpretação depende amplamente da natureza da aplicação do SP.

Interpretador

O interpretador pode ser visto na sua forma mais simples como um processo que executa o seguinte ciclo :

Verificação ou "Match": verifica quais produções da *PM* tem todas as suas condições satisfeitas pelo estado atual da base de dados. Isto é feito pela comparação dos *LHSs* de todas as produções da *PM* com o conteúdo da *WM*. O resultado dessa operação é um conjunto chamado conjunto de conflito *("conflit set")*. Se não houver nenhuma produção selecionada o processamento termina com *"fracasso"*.

Seleção: seleciona dentre as produções contidas no conjunto de conflito uma para execução.

Execução: executa as ações indicadas na *RHS* da produção selecionada do conjunto de conflito. Essas ações podem modificar o conteúdo da *MW*.

Teste: verifica se o objetivo do sistema já foi atingido; em caso negativo volta ao inicio do procedimento de verificação; em caso positivo termina o processamento com *"sucesso"*.

V.2.2 Estratégia de Controle nos *SP's*

Anteriormente vimos uma descrição breve de um ciclo realizado pelo interpretador de um *SP* genérico. Este comportamento dinâmico do *SP* é controlado por uma estratégia, isto é, temos que ter um meio de decidir qual a regra a ser aplicada em determinada situação.

A estratégia de controle pode ser implementada de diversas maneiras:

- dirigida por dados *("data-driven mode")* - a escolha da regra se baseia no estado observado da *WM*, só então uma ação correspondente é executada.
- propagação regressiva *("back chaining")* - a escolha da regra se baseia no objetivo de se conseguir um determinado estado presente na *WM*.
- uma combinação das duas anteriores.

Dependendo das características do domínio de aplicação uma ou mais estratégias serão escolhidas. Temos que observar, que a escolha de uma determinada estratégia de controle terá um impacto crítico sobre o desempenho do sistema no tempo e na possibilidade da solução de um determinadado problema.

V.3 SP Conexionista Distribuído - *(DCPS)*

Os primeiros modelos conexionista explorados por (Rumelhart & McClelland, 1986) mostraram que muitos fenômenos cognitivos que parecem requerer regras de forma explícitas podem ser tratados de outra forma. Estes fenômenos podem ser tratados em modelos conexionistas que capturam as regularidades de um determinado domínio sem, contudo, tornar esta regularidades explícitas. Uma das argumentações dos autores do *DCPS* é que existem tarefas em que a necessidade da representação das regras da forma explícita é imperativa.Eles citam como exemplo, que uma pessoa pode considerar uma regra explícita como *a antes de b exceto depois c* e pode aplicar esta regra em casos relevantes.Assim, a pessoa estará ciente da aplicação desta regra (quando ela estiver sendo aplicada, naturalmente) e não poderá aplicar um grande

número de regras diferentes em paralelo, embora possa ser capaz de realizar uma busca em paralelo por uma regra apropriada.

O *DCPS* é um modelo conexionista que pretende modelar, através de uma classe restrita dos sistemas de produção, o processo de representar e manipular estruturas simbólicas. Estas estruturas simbólicas são representadas por padrões de ativação na memória de trabalho e as regras são distribuídas sobre múltiplas conexões. Há um processo de verificação (*"match"*) e de ligação de variáveis onde operações usando estas variáveis são realizadas sequencialmente, tais como adicionar e retirar estruturas simbólicas da base de dados.

Basicamente o *DCPS* é constituido de cinco estruturas :

- *memória de trabalho,*
- *espaço de regras,*
- *espaço de ligação,*
- *espaço de cláusula 1,*
- *espaço de cláusula 2,*

que serão detalhadas posteriormente.

Como dissemos, o *DCPS* trabalha como várias restrições sobre o conceito geral de sistemas de produção visando simplificar a tarefa de modelagem e implementação. O *SP* em questão consiste de uma memória de trabalho que contém triplas de símbolos e um conjunto de produções que referenciam esta memória.

O *DCPS* utiliza um alfabeto de 25 símbolos constituído das letras de *A* até *Y*. A partir destes símbolos básicos teremos as triplas que são os componentes que serão efetivamente manipuladas pelas regras, isto é, serão adicionadas ou retiradas da memória de trabalho.

Quanto às regras, estas apresentam a seguinte estrutura: cada regra apresenta no lado esquerdo *(LHS)* duas triplas que deverão ser casadas com

a memória de trabalho a fim de adicionar ou retirar qualquer número de triplas especificado pelo lado direito da regra *(RHS)*, no caso da regra ser disparada. As variáveis podem aparecer em qualquer um dos lados da regra, sendo que no lado esquerdo representam uma restrição ao processo de *"match"* enquanto no lado direito,ao serem instanciadas, definem as ações que serão tomadas quando a regra for disparada.

Vejamos agoras algum exemplos das regras manipuladas pelo *DCPS*:

Regra 1 : $(ADY)(FKK) \longrightarrow +(GHJ)$

Regra 2 : $(ADY)(FKK) \longrightarrow -(ACR)$

Regra 3 : $(ADY)(FKK) \longrightarrow +(GHJ) - (ACR)$

Regra 4 : $(= xGH)(= xDD) \longrightarrow +(= xGJ)$

Regra 5 : $(= xKl)(= xQY) \longrightarrow -(= xC = x)$

Regra 6 : $(= xAB)(= xCD) \longrightarrow +(= xEF) + (PDQ) - (= xST)$

Regra 7 : $(= xFE)(= xDE) \longrightarrow -(E = xF) - (= xF = x)$

Desta forma, na regra 1 temos uma regra que não possui variáveis e quando a triplas (ADY) e (FKK) estiverem presente na memória de trabalho a regra será disparada ocasionando a inclusão da tripla (GHJ). Na regra 2 o processo é o mesmo da regra 1 porém ocorre a retirada da tripla (ACR). A regra 3 é uma composição das regras 1 e 2,tendo assim o mesmo efeito da regra 1 e da regra 2. A regra 4 já apresenta a variável x, que deverá está sempre na primeira posição quando no lado esquerdo da regra, podendo estar em qualquer posição do lado direito. A regra 4 nos diz que quando tivermos,por exemplo, triplas do tipo (AGH) e (ADD) ou (FGH) e (FDD) será adicionada à memória de trabalho a tripla (AGJ) ou (FGJ), respectivamente. As regras 5, 6 e 7 são outros exemplos das combinações possíveis das regras na sua forma mais generalizada.

V.4 A Arquitetura do Interpretador do *DCPS*

O interpretador do *DCPS* é composto de 5 conjuntos de unidades denominados *espaços*. O espaço denominado *memória de trabalho (WM)* fornece a entrada para dois *espaços de cláusula* rotulados de *C1* e *C2* respectivamente. Os espaços *C1* e *C2* são influenciados e influenciam dois outros espaços, denominados de *espaço de regras*, onde estão representadas as produções, e *espaço de ligação*, que é responsável pela ligação das variáveis.

V.4.1 Memória de Trabalho

A memória de trabalho do *DCPS* é um espaço constituido de 2000 unidades,onde cada unidade está associada a uma tabela chamada *campo receptivo*. Desta maneira uma unidade u_j é parte do conjunto de unidades u_i que representam uma tripla T_k se e somente se, sua tabela cr_j tem o primeiro, segundo e terceiro elementos da tripla na primeira, segunda e terceira coluna, respectivamente.

No modelo proposto em [TOUR88] observamos que somente teremos presente na memória de trabalho um número que varia de 6 a 24 triplas em um determinado instante, o que caracteriza a *MW* como uma matriz esparsa. Devido está característica da *WM* a representação distribuída é extremamente adequada pois diminui o número de unidades necessárias para representar uma tripla e acrescenta uma série de caracteristicas inerentes a este tipo de representação, como por exemplo características de tolerância a falhas, entre outras.

O campo receptivo de uma unidade é definido como sendo o conjunto de triplas geradas pelo produto vetorial de 6 símbolos em cada uma das 3 colunas, totalizando 216 triplas por campo.Podemos notar que cada receptor cobre aproximadamente:

$$\frac{totaldetriplasdeumreceptor}{totaldetriplaspossiveis} = \frac{216}{15625} = 0,01382 \cong 1,4\%$$

do espaço de todas as triplas possíveis.

Temos também que uma unidade pode participar de 6^3,isto é, 216

triplas e a memória de trabalho é constituida de 2000 unidades fornecendo 432000 possíveis triplas. Mas sabemos que com 25 símbolos, que podem ocupar 3 posições, nos teremos 15625 combinações, isto é, 15625 triplas. Como temos 432000 triplas possíveis e 15625 triplas possíveis e efetivas, concluimos que cada tripla da memória de trabalho é representada pela ativação de :

$$\frac{43200}{15625} = 27,65 \cong 28 unidades$$

C	A	B
F	E	D
M	H	J
Q	K	M
S	T	P
W	Y	R

Tabela V.1: Tabela de Campo Receptivo

A tabela V.1 é um exemplo de uma tabela de um campo receptivo gerado randomicamente [TOUR88] que está associado a uma unidade que junta com,aproximadamente, 27 outras unidades e participará da representação de 216 triplas dentre as quais, temos, (CAB), (CEB),(FKP), etc.

V.4.2 Espaço de Regras

Cada regra é representada pelo conjunto de 40 unidades e a ligação entre as unidades do espaço de regra com os espaços de cláusulas é determinado pelo lado esquerdo da regra. As unidades do espaço de regras têm conexões excitatórias bidirecionais entre o espaço de cláusula C1 e o espaço de cláusula C2. Quando as unidades de C1 e C2 se tornam ativas, as unidades do espaço de regras serão ativadas e estas unidades reforçarão as unidades ativas dos espaços de cláusula, já que as conexões são bidirecionais e excitatórias.

As 40 unidades que representam uma produção formam um pequeno circuito ou *'clique'*. Cada unidade ativa de um *'clique'* provê um pequeno estímulo excitatório para as outras unidades do mesmo *'clique'* e um outro pequeno estímulo

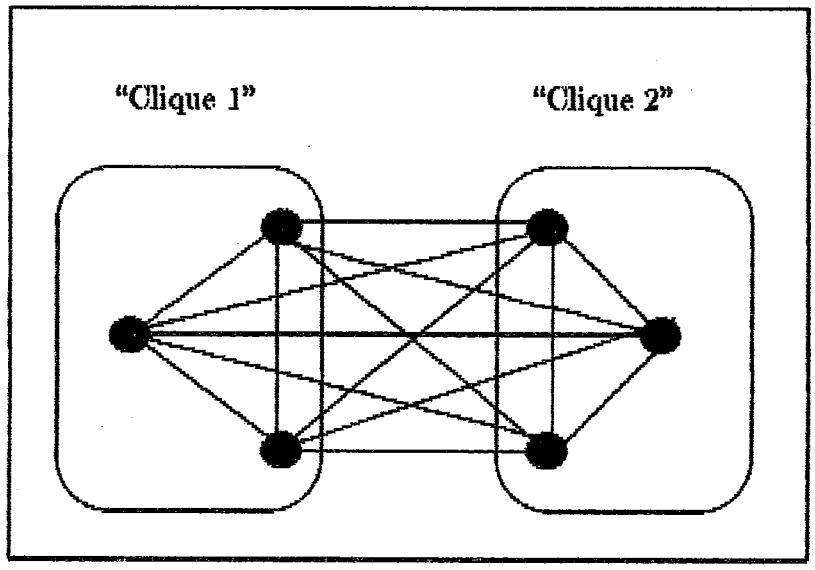

Figura V.1: Rede "winner-take-all" composta de dois "cliques" com três unidades, onde cada um requer $\frac{(N^2-N)}{2}$ conexões

inibitório para todas as outras unidades dos outros *'cliques'* (veja figura V.2).Na realidade o que temos é uma rede do tipo *'winner-take-all'* de forma que quando a rede estabiliza todas as unidades de um *'clique'* estarão ativas enquanto as unidades pertencentes aos outros *'cliques'* estarão inativas. Desta maneira o sistema decide qual regra será disparada.

V.4.3 Espaços de Cláusulas

Os espaços de cláusula, rotulados de $C1$ e $C2$, têm por função realçar determinadas triplas. Podemos explicar sua função em termos de uma analogia com o processo humano de percepção. Se consideramos um observador em um ambiente, como por exemplo uma sala, e por qualquer motivo o observador fixar a atenção em um dos objetos desta sala, então esse processo de focalizar a atenção nesse objeto é análogo, em função, ao objetivo dos espaços de cláusulas.

Já vimos que uma das restrições do *DCPS* é permitir somente duas triplas no lado esquerdo de qualquer produção. Este fato está intimamente relacionado aos espaços de cláusulas, pois cada um deles é responsável por uma das duas triplas. Os espaços de clásulas, $C1$ e $C2$, permite que triplas específicas presente na memória de trabalho sejam 'escolhidas' a fim de realizar o *"match"* com as cláusulas

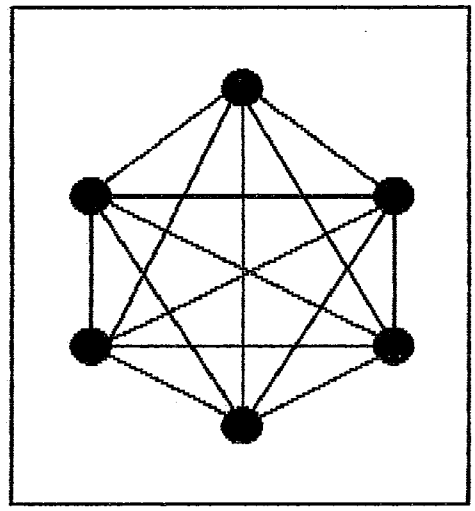

Figura V.2: Espaço de unidades com $\frac{(N^2-N)}{2}$ conexões bidirecionais

no espaço das regras.

Cada espaço de cláusula contém 2000 unidades que mantém uma relação unívoca com as unidades da memória de trabalho. Cada unidade da *MW* mantém uma conexão excitatória com as unidades correspondentes em *C1* e *C2*, ocasionando um reflexo das ativações da *MW* nas unidades de *C1* e *C2* As unidades do espaço de cláusula possuem ligações inibitórias e são projetadas para permitir que ocorra a ativação de somente 28 unidades por espaço, isto é, somente uma tripla estará presente em cada um dos espaços *C1* e *C2*.

Um problema que poderia ocorrer é a presença de triplas nos espaços *C1* e *C2* que não mantivessem nenhuma relação com as triplas presentes na *MW*. Mas este fato não deverá ser mantido por muito tempo, caso ocorra, pois, vários parâmetros da rede só permitem que uma tripla se mantenha no espaço de cláusula se tiverem um suporte das unidades do espaço de regras e do espaço de ligação, bem como das unidades da memória de trabalho.

Como dissemos anteriormente, existe um mapeamento, através de conexões excitatórias, entre as unidades da *MW* e os espaços *C1* e *C2*. Temos que a topologia do espaço de cláusula requer $\frac{(N^2-N)}{2}$ conexões inibitórias bidirecionais o que ocasiona o crescimento extremamente rápido das conexões conforme o aumento do número de unidades (veja figura V.3). A solução adotada pelos autores foi

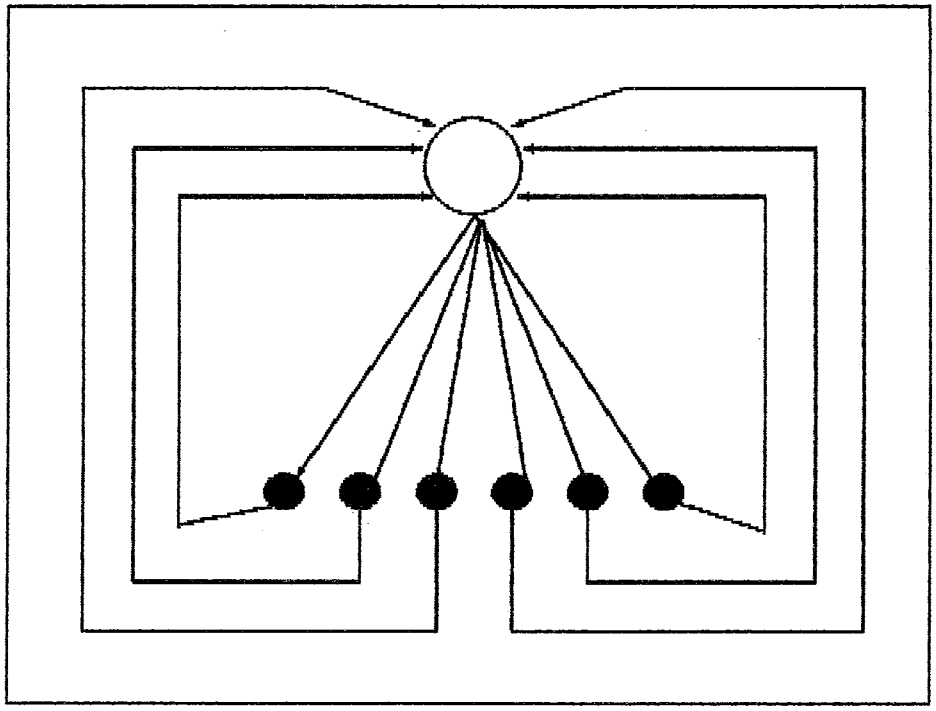

Figura V.3: Unidade Reguladora

a seguinte: utilizar N conexões excitatórias das unidades do espaço de cláusula para uma unidade reguladora e N conexões inibitórias na direção oposta, isto é, da unidade reguladora para as unidades do espaço de cláusula e uma ligação de cada unidade para ela mesma, desta maneira o total de conexões passa de $\frac{(N^2-N)}{2}$ para $3N$ (veja figura V.4) . Porém para as análises teóricas os autores não consideram este artifício, mantendo a topologia original.

V.4.4 Espaço de Ligação

As unidades deste espaço constituem uma rede do tipo *'winner-take-all'* com 25 *cliques*, sendo um *clique* para cada símbolo do alfabeto. Este espaço utiliza a representação distribuída em que cada unidade pertence a três *cliques*, em outras palavras, cada unidade participa da representação de 3 símbolos. São utilizadas 40 unidades do espaço de ligação para representar cada símbolo, assim temos:

$$\frac{25\times40}{3} = \frac{1000}{3} \cong 333 \text{ unidades}$$

como o número total de unidades do espaço de ligação. Desta forma temos 40 unidades para cada símbolo, com exceção do último símbolo representado por 39 unidades.

Cada unidade deste espaço tem um conjunto de conexões bidirecionais excitatórias que se ligam com todas as unidades dos espaços *C1* e *C2* que contém o campo receptivo associado ao símbolo que esta unidade contribui na representação.

Basicamente, o que nos interessa é ressaltar o tipo de representação distribuída utilizada pelo DCPS, isto é, a representação distribuída com a técnica "coarse-coded". Como vimos no capítulo III, esta técnica utiliza um conjunto arbitrário de unidades formando o que chamamos de padrão, e as unidades não possuem, desta forma, nenhuma relação semântica com o símbolo representado. Uma das grandes vantagens desta técnica é a eficiência com que manipula memórias esparsas, apesar da necessidade de mecanismos complexos de acesso. Sendo assim, veja [TOUR88] para maiores detalhes do funcionamento do DCPS, já que tal questão foge ao escopo deste trabalho.

Uma evolução deste modelo é o BoltzCONS, [TOUR90], modelo baseado no DCPS que cria e manipula dinamicamente estruturas de símbolos, como pilhas e árvores, de forma análoga a listas ligadas manipuladas pela linguagem LISP.

Capítulo VI

Representação por Produto Tensorial

VI.1 Introdução

Sabemos que a representação de estruturas simbólicas é de suma importância para o tratamento de uma classe de problemas em Inteligência Artificial pela ótica conexionista. Uma maneira inicial de se abordar este problema da representação estruturada em sistemas conexionistas é nos limitarmos à investigação de como podemos tratar o problema das ligação de variáveis.

Estudaremos neste capítulo a proposta de Smolesnky ([SMOL87], [SMOL87b], [SMOL88], [SMOL89], [SMOL90]) da *Representação por Produto Tensorial* que tem como objetivo fornecer um *"framework"* para análise matemática de diversos modelos e propostas de solução do problema de ligação.

VI.1.1 O que é Representação por Produto Tensorial

Basicamente a Representação por Produto Tensorial (*RPT*) é uma técnica desenvolvida por Paul Smolensky que estabelece uma formalização da idéia que um conjunto de pares variável/valor pode ser representado pela acumulação da atividade em um conjunto de unidades em que cada unidade computa o produto entre uma característica de uma variável e uma característica de seu valor. Está técnica objetiva, assim, a representação completamente distribuída de ligações e estruturas simbólicas

onde os outros tipos de representação (local e parcialmente distribuída) são casos particulares do caso geral.

Em termos de técnica de representação de estruturas simbólicas em redes conexionistas, a representação por produto tensorial é considerada a mais complexa e a que apresenta maior grau de formalização [GELD90]. Na realidade o objetivo da técnica pode ser resumido em achar um mapeamento entre um conjunto de objetos estruturados (árvores, listas, por exemplo) e um espaço vetorial, de tal forma que várias relações entre os constituintes da estrutura sejam preservadas e que a representação da estrutura, como um todo, seja resultante da representação das partes.

Segundo [GELD90] a *RPT* foi resultado de dois *"insight"* de Smolensky:

1. Smolensky ampliou o conceito que, sob determinadas condições, a adição de vetores pode ser utilizada para combinar duas representações a fim de gerar uma representação de um objeto mais complexo. Assim, Smolensky utiliza uma operação mais complexa (produto tensorial), porém, muito mais poderosa para trabalhar o conceito de formação de representações complexas a partir de representações mais simples. Além disso, o produto tensorial produz um vetor que pode ser utilizado em somas vetoriais ou produtos tensoriais como qualquer outro vetor (veremos isso mais adiante).

2. O segundo *"insight"* de Smolensky é a introdução da noção de *decomposição em papéis* (*"role decomposition"*) que é um método geral para a redução de itens complexos estruturados em conjuntos de pares, isto é, um item complexo a ser representado é analisado como sendo constituído de um conjunto de *papéis* (*"roles"*) com cada papel tendo um certo preenchedor (veremos isso também, com maiores detalhes, mais adiante).

VI.1.2 Características da Representação por Produto Tensorial

A Representação por Produto Tensorial apresenta uma série de características e prováveis propriedades que a tornam uma candidata a uma solução geral para o problema da representação conexionista de estruturas simbólicas. Como o estudo das bases teóricas deste tipo de representação ainda está em fase inicial de desenvolvimento, se faz mister a elaboração de elos entre diversas áreas teóricas para a elucidação de diversos problemas e direcionamento de soluções.

Vejamos agora algumas características da *RPT*:

- A representação completamente distribuída de estruturas simbólicas é formada a partir da representação distribuída das partes constituintes da estrutura.
- Quase todas as abordagens de representação conexionista de estruturas são casos particulares deste método.
- Cada componente constituinte da estrutura pode ser extraido com completa precisão se a estrutura representada não ultrapassar os limires de saturação da rede.
- Estruturas de tamanho ilimitado podem ser representadas em uma rede limitada com uma degradação suave da representação.
- A representação se aplica tanto a estruturas contínuas como discretas.
- O mecanismo de ligação por produto tensorial é realizado de forma simples em uma rede neuronal.
- A representação respeita a independência de dois aspectos do paralelismo na ligação de variável:
  - geração de ligação
  - manutenção de ligação
- Os constituintes das estruturas podem ser facilmente extraidos na mesma rede que realiza a ligação.

- O produto tensorial permite que um mesmo padrão de atividade que representa uma variável seja representante de um valor.
- A representação pode ser usada recursivamente .
- Representações conexionistas de operações sobre estruturas simbólicas e tipos de dados recursivos podem ser analisadas naturalmente.
- A recuperação de representações de dados estruturados em memórias conexionistas pode ser formalmente analisada e em alguns casos estabelecendo-se condições necessárias e suficientes para que o conjunto de estruturas armazenadas não sofram interferência na memória.

VI.2 Álgebra Tensorial

Nesta seção estudaremos alguns tópicos da álgebra tensorial que usaremos na análise e formalização de vários conceitos e propriedades da *RPT*. Desta maneira, delinearemos, somente, a estrutura básica indicando os seguintes autores para um estudo mais elaborado: [LOOM68], [SATA75], [GREU81], [GREU78], [SHAW82].

A teoria do Produto Tensorial é parte integrante da álgebra linear e interage com diversos outros ramos da matemática, especialmente com a teoria da representação de grupos. Uma das principais caracteríristicas do Produto Tensorial é combinar a abordagem contínua, da teoria dos espaços vetoriais, com a abordagen discreta, da teoria dos grupos, em uma única teoria. Tem sido usada em diversos ramos da física, com destaque na mecânica quântica.

Podemos definir o produto tensorial de duas maneiras: com respeito a uma base e , independente da escolha de qualquer base.

A definição de produto tensorial com respeito a uma base é a mais simples e mais divulgada. Consideremos dois espaços vetoriais $\mathcal{U}$ e $\mathcal{V}$ com dimensões n e m e bases $\{\hat{u}_i\}$ e $\{\hat{v}_j\}$. O produto tensorial toma um vetor u de $\mathcal{U}$ e um vetor v de $\mathcal{V}$ e gera um vetor pertencente a $\mathcal{U} \otimes \mathcal{V}$. A dimensão de $\mathcal{U} \otimes \mathcal{V}$ é igual a $m \times n$, os vetores $\{\hat{u}_i \otimes \hat{v}_j\}$ formam uma base para o espaço vetorial $\mathcal{U} \otimes \mathcal{V}$. Se os componentes

de $u \in \mathcal{U}$ com respeito à base $\{\hat{u}_i\}$ são $\{u_i\}$ e os componentes de $v \in \mathcal{V}$ com respeito à base $\{\hat{v}_j\}$ são $\{v_j\}$ então os componentes de $\mathcal{U} \otimes \mathcal{V}$ com respeito à base $\hat{u}_i \otimes \hat{v}_j$ são $u_i v_j$.

A forma de se definir o produto tensorial de maneira independente de uma base não é tão imediata e envolve, segundo Smolensky, grandes sutilezas, porém, fornece um entendimento mais profundo das operações das quais manifestações particulares podem ser derivadas ao invés de estipuladas [SMOL87]. Para o desenvolvimento da definição de produto tensorial independente de uma base são requeridas três fases (veja figura VI.1):

1. Conceito de dualidade entre espaços vetoriais e certos espaços de funções.
2. Definição de produto tensorial para espaço de funções.
3. Transferência da definição de produto tensorial do espaço de funções para os espaços vetorias originais pela dualidade.

VI.3 Definição de Produto Tensorial relativo a uma Base

Definição VI.3.1 *Sejam $\mathcal{U}$ e $\mathcal{V}$ dois espaços vetoriais de dimensões m e n respectivamente. Chamamos de Produto Tensorial de $\mathcal{U}$ por $\mathcal{V}$ a todo par $(\mathcal{Z}, \Phi)$ que satisfaça aos seguintes axiomas:*

1. $\mathcal{Z}$ é um espaço vetorial e $\Phi : \mathcal{U} \times \mathcal{V} \longrightarrow \mathcal{Z}$ é uma aplicação bilinear do par $\mathcal{U}$, $\mathcal{V}$ em $\mathcal{Z}$;
2. $dim\mathcal{Z} = dim\mathcal{U} dim\mathcal{V}$;
3. $\Phi(\mathcal{U} \times \mathcal{V})$ gera $\mathcal{Z}$.

Comentário 1:

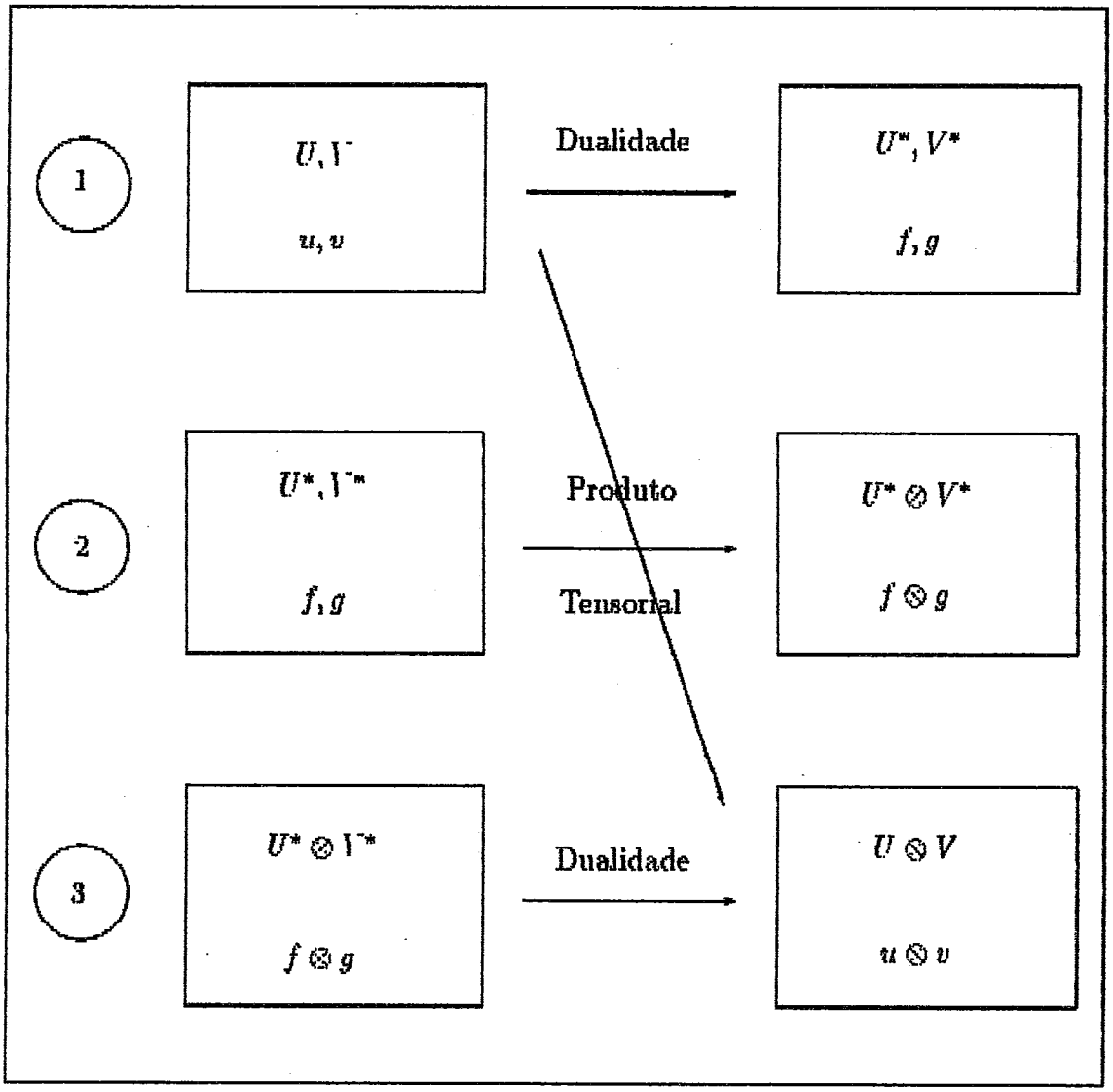

Figura VI.1: Definição de produto tensorial independente de uma base

Indicaremos quando for mais conveniente um produto tensorial de $\mathcal{U}$ por $\mathcal{V}$ por $\mathcal{U} \otimes \mathcal{V}$ e $\Phi(u,v)$ por $u \otimes v$

Comentário 2:

Os axiomas 2 e 3 são equivalentes a um único axioma (2') que é o seguinte: Se $(\hat{u}_i)$ e $(\hat{v}_j)$ são as respectivas bases de $\mathcal{U}$ e $\mathcal{V}$ para $1 \leq i \leq m$ e $1 \leq j \leq n$, então $(\Phi(\hat{u}_i, \hat{v}_j))$ é uma base para $\mathcal{Z}$.

Prova (se axiomas 2 e 3 então axioma 2'):

Dados $u = \sum \alpha_i \hat{u}_i \in \mathcal{U}$ e $v = \sum \beta_j \hat{v}_j \in \mathcal{V}$ e sendo Φ bilinear temos:

$\Phi(u,v) =$

$\Phi(\sum \alpha_i \hat{u}_i, v) =$

$\Phi(\alpha_1 \hat{u}_1 + \cdots + \alpha_m \hat{u}_m, v) =$

$\Phi(\alpha_1 \hat{u}_1, v) + \cdots + \Phi(\alpha_m \hat{u}_m, v) =$

$\alpha_1 \Phi(\hat{u}_1, v) + \cdots + \alpha_m \Phi(\hat{u}_m, v) =$

$\sum_1^m \alpha_i \Phi(\hat{u}_i, v) =$

$\sum_1^m \alpha_i \Phi(\hat{u}_i, \beta_1 \hat{v}_1 + \cdots + \beta_n \hat{v}_n) =$

$\sum_1^m \alpha_i \Phi(\hat{u}_i, \beta_1 \hat{v}_1) + \cdots + \sum_1^m \alpha_i \Phi(\hat{u}_i, \beta_n \hat{v}_n) =$

$\sum_1^m \alpha_i \beta_1 \Phi(\hat{u}_i, \hat{v}_1) + \cdots + \sum_1^m \alpha_i \beta_n \Phi(\hat{u}_i, \hat{v}_n) =$

$\sum_1^m \alpha_i \sum_1^n \beta_j \Phi(\hat{u}_i, \hat{v}_j) =$

$\sum_{i,j} \alpha_i \beta_j \Phi(\hat{u}_i, \hat{v}_j), onde (1 \leq i \leq m, 1 \leq j \leq n).$

Assim, para todo $u \in U$ e todo $v \in \mathcal{V}$, $\Phi(u,v)$ pode ser expresso como uma combinação linear dos elementos $\Phi(\hat{u}_i, \hat{v}_j)$. Como temos que $\Phi(\mathcal{U} \times \mathcal{V})$ gera $\mathcal{Z}$ (axioma 3), então a mn-upla $\Phi(\hat{u}_i, \hat{v}_j)$ também gera $\mathcal{Z}$. Temos também que $dim\mathcal{Z} = mn$ (axioma 2), daí a mn-upla $\Phi(\hat{u}_i, \hat{v}_j)$ é uma base de $\mathcal{Z}$. Logo o axioma 2' é verdadeiro.

Prova (se axiomas 2' então axiomas 2 e 3): Trivial.

Veremos agora que dados dois espaços vetoriais sempre existe um outro espaço vetorial e uma função bilinear que satisfaz os axiomas do produto tensorial. E mais, este par (espaço vetorial,função bilinear) é único a menos de um

isomorfismo canônico.

Teorema VI.3.1 *(Teorema da existência) Dados dois espaços vetoriais $\mathcal{U}$ e $\mathcal{V}$, sempre* existe *um espaço vetorial $\mathcal{Z}$ e uma função bilinear $\Phi : \mathcal{U} \times \mathcal{V} \longrightarrow \mathcal{Z}$ que satisfaz aos axiomas do produto tensorial.*

Prova:(Por construção)

Seja $\mathcal{Z}$ um espaço vetorial qualquer com dimensão mn e $(\hat{z}_{ij})$ $(1 \leq i \leq m\ ,\ 1 \leq j \leq n)$ uma base para $\mathcal{Z}$. Seja também $(\hat{u}_i)$ $(\ 1 \leq i \leq m)$ base para $\mathcal{U}$ e $(\hat{v}_j)$ $(1 \leq j \leq n\ )$ base para $\mathcal{V}$ e $v = \sum_i \alpha \hat{u}_i$ e $u = \sum_j \beta \hat{v}_j$ para todo $u \in \mathcal{U}$ e $v \in \mathcal{V}$ e $\alpha\ ,\ \beta \in \Re$. Podemos definir uma função bilinear $\Phi : \mathcal{U} \times \mathcal{V} \longrightarrow \mathcal{Z}$ tal que:

$$\Phi(u, v) = \sum_{i,j} \alpha_i \beta_j \hat{z}_{ij}$$

Pelo comentário da definição VI.3.1 temos que o par $(\mathcal{Z},\ \Phi)$ satisfaz aos axiomas do Produto Tensorial.

Teorema VI.3.2 *(Propriedade Universal) Sejam $\mathcal{U}$, $\mathcal{V}$, $\mathcal{Z}$ espaços vetoriais e $\Phi : \mathcal{U} \times \mathcal{V} \longrightarrow \mathcal{Z}$ tal que $\mathcal{U} \otimes \mathcal{V}$ esteja definido. Para qualquer par $(\mathcal{Z}', \oplus')$ onde $\mathcal{Z}'$ é um espaço vetorial e $\Phi' : \mathcal{U} \times \mathcal{V} \longrightarrow \mathcal{Z}'$, existe uma única função linear $\rho : \mathcal{Z} \longrightarrow \mathcal{Z}'$ tal que:*

$$\Phi'(u, v) = \rho(\Phi(u, v)),\ \textit{para todo}\ u \in \mathcal{U}\ \textit{e todo}\ v \in \mathcal{V}.$$

Prova :

Dado uma função bilinear $\Phi' : \mathcal{U} \times \mathcal{V} \longrightarrow \mathcal{Z}'$, podemos definir uma função linear $\rho : \mathcal{Z} \longrightarrow \mathcal{Z}'$ por:

$$\rho(\textstyle\sum_{i,j} \alpha_{ij} \hat{z}_{ij}) = \sum_{i,j} \alpha_{ij} \Phi'(\hat{u}_i, \hat{v}_j)$$

daí e do teorema da existência:

$$\rho(\Phi(u, v)) = \rho(\textstyle\sum_{i,j} \alpha_i \beta_j \hat{z}_{ij}) = \sum_{i,j} \alpha_i \beta_j \Phi'(\hat{u}_i, \hat{v}_j) = \Phi'(u, v).$$

Reciprocamente, se ρ é uma função linear que satisfaz a :

$$\Phi'(u,v) = \rho(\Phi(u,v)), \text{ para todo } u \in U \text{ e todo } v \in \mathcal{V}.$$

então,

$$\rho(\hat{z}_{ij}) = \rho(\Phi(\hat{u}_i, \hat{v}_j)) = \Phi'(\hat{u}_i, \hat{v}_j)$$

daí segue que:

$$\rho(\sum_{ij} \alpha_{ij}\hat{z}_{ij}) = \sum_{i,j} \Phi'(\hat{u}_i, \hat{v}_j).$$

Assim a função linear que satisfaz à $\Phi'(u,v) = \rho(\Phi(u,v))$, para todo $u \in U$ e todo $v \in \mathcal{V}$ é unicamente determinada.

Teorema VI.3.3 *Sejam $\mathcal{U} \otimes \mathcal{V}$ e $\mathcal{U} \tilde{\otimes} \mathcal{V}$ dois produtos tensoriais de $\mathcal{U}$ por $\mathcal{V}$. Existe um único isomorfismo canônico de $\mathcal{U} \otimes \mathcal{V}$ sobre $\mathcal{U} \tilde{\otimes} \mathcal{V}$ que leva $u \otimes v$ em $u \tilde{\otimes} v$, para todo $u \in \mathcal{V}$ e $v \in \mathcal{V}$.*

Prova :

Vamos supor que existem duas funções bilineares tal que:

$$\Phi : \mathcal{U} \times \mathcal{V} \longrightarrow \mathcal{U} \otimes \mathcal{V}$$
$$\Phi' : \mathcal{U} \times \mathcal{V} \longrightarrow \mathcal{U} \tilde{\otimes} \mathcal{V}$$

satisfazem a definição VI.3.1. Do teorema VI.3.2 temos que existe uma única função linear tal que:

$$\rho : \mathcal{U} \otimes \mathcal{V} \longrightarrow \mathcal{U} \tilde{\otimes} \mathcal{V}$$

e existe também outra função linear (também única) tal que:

$$\rho' : \mathcal{U} \otimes \mathcal{V} \longrightarrow \mathcal{U} \tilde{\otimes} \mathcal{V}$$

satisfazendo $\Phi'(u,v) = \rho(\Phi(u,v))$, para todo $u \in U$ e todo $v \in \mathcal{V}$. Temos assim que:

$$\rho'(\rho(\Phi(u,v))) = \rho'(\Phi'(u,v)) = \Phi(u,v).$$

Como temos que $\mathcal{U} \otimes \mathcal{V}$ é gerado por $\Phi(\mathcal{U} \times \mathcal{V})$ (definição VI.3.1), então:

$$\rho' \circ \rho = 1_{\mathcal{U}\otimes\mathcal{V}}$$

onde $1_{\mathcal{U}\otimes\mathcal{V}}$ denota a transformação identidade do espaço $\mathcal{U} \otimes \mathcal{V}$. Similarmente:

$$\rho \circ \rho' = 1_{\mathcal{U}\otimes\mathcal{V}}.$$

Concluimos, assim, que ρ e ρ' são isomorfismos, inverso um do outro, isto é,

$$\rho' = \rho^{-1}$$

A unicidade de ρ é resultado de ser sua ação definida num conjunto de geradores de $\mathcal{U} \otimes \mathcal{V}$.

VI.4 Definição de Produto Tensorial independente de uma Base

VI.4.1 Espaço Dual

Definição VI.4.1 *Seja $\mathcal{V}$ um espaço vetorial e $\mathcal{K}$ um corpo. Uma função $f : \mathcal{V} \longrightarrow \mathcal{K}$ satisfazendo às condições:*

1. $f(v+u) = f(v) + f(u)$, $v,u \in \mathcal{V}$
2. $f(\alpha v) = \alpha f(v)$, $u \in \mathcal{V}$ *e* $\alpha \in \mathcal{K}$

é denominada de funcional linear ou forma linear sobre o espaço vetorial $\mathcal{V}$. Denominaremos por $\mathcal{V}^*$ o conjunto de todas as aplicações lineares de $\mathcal{V}$ em $\Re$, isto é, o conjunto dos funcionais lineares sobre o espaço vetorial $\mathcal{V}$. $\mathcal{V}^*$ é chamado de espaço dual.

Lema VI.4.1 *Seja* $\{\vec{v}_1, \cdots, \vec{v}_n\}$ *uma base do espaço vetorial* $\mathcal{V}$. *Um funcional linear* $f \in \mathcal{V}^*$ *fica determinado quando conhecemos os* n *números:* $f(\vec{v}_1) = \alpha_1, \cdots, f(\vec{v}_n) = \alpha_n$.

Prova :

Temos que para qualquer $v \in \mathcal{V}$, $v = \sum \xi_i \vec{v}_i$, onde $\xi_1, \cdots, \xi_n$ são as coordenadas de v em relação à base ordenada $\{\vec{v}_1, \cdots, \vec{v}_n\}$.

Daí,

$$f(v) = f(\sum \xi_i \vec{v}_i) = \sum_i \xi_i f(\vec{v}_i) = \sum_i \xi_i \alpha_i.$$

Lema VI.4.2 *Seja* $\mathcal{E} = \{\vec{v}_1, \cdots, \vec{v}_n\}$ *uma base de* $\mathcal{V}$. *Pelo lema anterior podemos definir* n *funcionais lineares* $\vec{v}_i^* : \mathcal{V} \longrightarrow \Re$ *por:*

$$\vec{v}_i^*(\vec{v}_j) = \delta_{ij}$$

onde,

$$\delta_{ij} = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$$

onde $i, j = 1, \ldots, n$ *e* $\vec{v}_j \in \mathcal{E}$.

Então, se $\mathcal{E}^* = \{\vec{v}_1^*, \cdots, \vec{v}_n^*\}$ *é uma base de* $\mathcal{V}^*$ *e dado qualquer* $f \in \mathcal{V}^*$, *suas componentes* (f_i) *com respeito à base* $\mathcal{E} = \{\vec{v}_1, \cdots, \vec{v}_n\}$ *são:*

$$f_i = f(\vec{v}_i).$$

Prova :

Seja $\vec{v} \in \mathcal{V}$, logo $\vec{v}$ pode ser escrito como uma combinação linear dos elementos da base $\mathcal{E}$:

$$\vec{v} = \sum_i v_i \vec{v}_i$$

temos, também, que:

$$\hat{v}_i^*(\vec{v}) = \hat{v}_i^*(\sum_j v_j \vec{\hat{v}}_j) =$$
$$= \sum v_j \hat{v}_i^*(\vec{\hat{v}}_j) = \sum v_j \delta_{ij} = v_i.$$

Logo, $\hat{v}_i^*$ é o funcional linear que extrai o i-ésimo componente de um vetor qualquer pertencente ao espaço vetorial $\mathcal{V}$, com respeito a uma base $\mathcal{E}$.

Consideremos, agora, um funcional linear qualquer pertencente os espaço dual $\mathcal{V}^*$, isto é, $f \in \mathcal{V}^*$. Como f é um funcional linear, temos:

$$f(\vec{v}) = f(\sum_i v_i \vec{\hat{v}}_i) = \sum_i v_i f(\vec{\hat{v}}_i) =$$
$$= \sum \hat{v}_i^*(\vec{v}) = [\sum_i f_i \hat{v}_i^*](\vec{v}) =, \text{ para todo } v \in \mathcal{V}.$$

Como é verdadeiro para todo $\vec{v} \in \mathcal{V}$:

$$f = \sum_i f_i \hat{v}_i^*.$$

Daí, para todo $f \in \mathcal{V}^*$, temos que f pode ser escrito como uma combinação linear dos funcionais lineares $\{\hat{v}_i^*\}$, logo, $\mathcal{E}^*$ gera $\mathcal{V}^*$.

Assim, ainda nos resta mostrar que $\mathcal{E}^*$ é linearmente independente para que seja uma base para $\mathcal{V}^*$. Desta forma, suponhamos que:

$$\sum_i \alpha_i \hat{v}_i^* = 0$$

logo, para todo v pertencente ao espaço vetorial $\mathcal{V}$, temos:

$$0 = (\sum_i \alpha_i \hat{v}_i^*)(\vec{v}) = \sum_i \alpha_i \hat{v}_i^*(\vec{v}) = \sum_i \alpha_i v_i$$

Como a única maneira para que $\sum_i \alpha_i v_i = 0$ seja sempre verdadeira é que $\alpha_i = 0$, $i = 1, \ldots, n$, logo $\mathcal{E}^* = \{\hat{v}_i^*\}$, $i = 1, \ldots, n$. é linearmente independente.

Portanto, como $\mathcal{E}^*$ gera $\mathcal{V}^*$ e $\mathcal{E}^*$ é linearmente independente, $\mathcal{E}^*$ é uma base para $\mathcal{V}^*$.

Teorema VI.4.1 *Se $\mathcal{V}$ é um espaço vetorial e $\mathcal{V}^*$ seu dual , então $\mathcal{V}$ e $\mathcal{V}^*$ são espaços vetoriais isomórficos, embora não exista um isomorfismo canônico entre eles.*

Prova :

Como uma base $\mathcal{E}$ e sua dual $\mathcal{E}^*$ têm o mesmo número de elementos, a dimensão do espaço dual de um espaço vetorial $\mathcal{V}$ é igual à dimensão de $\mathcal{V}$, portanto $\mathcal{V}$ e $\mathcal{V}^*$ são isomorfos.

Se $\mathcal{E} = \{\hat{v}_1, \ldots, \hat{v}_n\}$ e $\mathcal{E}^* = \{\hat{v}_1^*, \ldots, \hat{v}_n^*\}$ é a base dual de $\mathcal{E}$, a aplicação $\mathcal{T}_\mathcal{E} : \mathcal{V} \longrightarrow \mathcal{V}^*$ que ao vetor $\vec{v} = \sum \alpha_i \hat{u}_i$ associa o funcional $\mathcal{T}_\mathcal{E}(\vec{v}) = \vec{v} = \sum \alpha_i v_i^*$ é um isomorfismo que depende da base $\{\hat{v}_1, \ldots, \hat{v}_n\}$, pois se tomássemos uma outra base $\mathcal{D} = \{\hat{d}_1, \ldots, \hat{d}_n\}$ a aplicação $\mathcal{T}_\mathcal{D} : \mathcal{V} \longrightarrow \mathcal{V}^*$ definida de maneira análoga a $\mathcal{T}_\mathcal{E}$ seria também um isomorfismo mais não não necessariamente o mesmo.

Teorema VI.4.2 *Existe um isomorfismo canónico $\mathcal{I}$ entre $\mathcal{V}$ e $\mathcal{V}^{**}$.*

Prova:

Seja $\mathcal{I}$ o funcional linear tal que

$$\mathcal{I}(\vec{v})(g) = g(\vec{v}), \vec{v} \in \mathcal{V}, g \in \mathcal{V}*.$$

A aplicação $\mathcal{I}$ é biunívoca, isto é, $\mathcal{I}(\vec{v}) = 0$ implica $\vec{v} = 0$. De fato, $\mathcal{I}(\vec{v}) = 0$ significa que $\mathcal{I}(\vec{v})(g) = g(\vec{v}) = 0$, qualquer que seja $g \in \mathcal{V}^*$. Seja $\mathcal{E} = \{\vec{\hat{v}}_1, \ldots, \vec{\hat{v}}_n\}$ uma base de $\mathcal{V}$, e suponhamos que $\vec{v} = \sum \alpha_i \vec{\hat{v}}_i$.

Então,

$$g(\vec{v}) = \sum \alpha_i g(\vec{\hat{v}}_i) = 0$$

qualquer que seja $g \in \mathcal{V}*$.

Considerando à base dual $\mathcal{E}^* = \{\hat{v}_1^*, \ldots, \hat{v}_n^*\}$ e tomando sucessivamente $g = \hat{v}_1^*, g = \hat{v}_2^*, \ldots, g = \hat{v}_n^*$, obteremos

$$\alpha_1 = 0, \alpha_2 = 0, \ldots, \alpha_n = 0,$$

donde $\vec{v} = 0$. Como $\mathcal{V}$ e $\mathcal{V}^{**}$ têm a mesma dimensão segue-se, que $\mathcal{I}$ é um isomorfismo.

Teorema VI.4.3 *Cada funcional da base dual $\hat{v}_i^* \in \mathcal{V}^*$ é equivalente ao produto interno com algum vetor $\vec{u}_i \in \mathcal{V}$:*

$$\hat{v}_i^*(\vec{v}) = \vec{u}_i.\vec{v}, \text{ para todo } \vec{v} \in \mathcal{V}.$$

Prova:

Seja V_i o subespaço de $\mathcal{V}$ gerado por todos os vetores da base a menos de $\hat{v}_1^*$:

$$V\_i = gerado(\{\vec{\hat{v}}^*_j \mid j \neq i\})$$

Temos que $\mathcal{V}_{-i}$ é um subespaço de dimensão $n-1$, já que V tem dimensão n. Existe um subespaço unidimensional ortogonal a $\mathcal{V}_{-i}$ e chama-lo-emos de $\mathcal{V}_i$. Seja w_i um vetor não nulo pertencente a $\mathcal{V}_i$. Definamos que:

$$\vec{u}_i = \frac{\vec{w}_i}{\vec{w}_i\vec{\hat{v}}^*_i}$$

logo,

$$\vec{u}_i\vec{\hat{v}}^*_i = \frac{\vec{w}_i\vec{\hat{v}}^*_i}{\vec{w}_i\vec{\hat{v}}^*_i} = 1$$

Se $j \neq i$ então $\vec{u}_i\vec{\hat{v}}^*_j = 0$, uma vez que $\vec{u}_i \in \mathcal{V}_i$, $\hat{v}_j^* \in \mathcal{V}_{-i}$, e $\mathcal{V}_i$ é ortogonal a $\mathcal{V}_{-i}$. Assim,

$$f_i : \mathcal{V} \longrightarrow \Re; \vec{v} \longmapsto \vec{u}_i\vec{v}$$

é um funcional linear com os seguintes valores:

$$f_i(\vec{\hat{v}}^*_j) = \delta_{ij}$$

Logo, $f_i = \hat{v}_i^*$

VI.4.2 Produto Tensorial de Funcionais

Definição VI.4.2 *Seja $f \in \mathcal{U}^*$ e $g \in \mathcal{V}^*$. O produto tensorial de f e g é o funcional $f \otimes g$ definido por : $f \otimes g : \mathcal{U} \times \mathcal{V} \longrightarrow \Re; (\vec{u}, \vec{v}) \longmapsto f(\vec{u})g(\vec{v})$.*

Comentário:

O produto tensorial usa a multiplicação em $\Re$ para combinar duas funções de uma variável em uma função de duas variáveis.

Proposição VI.4.1 *Seja a função produto tensorial $\pi = f \otimes g$, então as seguintes propriedades são válidas:*

1. $\pi(\alpha\vec{u}_1 + \beta\vec{u}_2, \vec{v}) = \alpha\pi(\vec{u}_1, \vec{v}) + \beta\pi(\vec{u}_2, \vec{v})$
2. $\pi(\vec{u}, \alpha\vec{v}_1 + \beta\vec{v}_2) = \alpha\pi(\vec{u}, \vec{v}_1) + \beta\pi(\vec{u}, \vec{v}_2)$

Prova:

Usando a propriedade da linearidade do funcional π temos:

$1)\pi(\alpha\vec{u}_1, \vec{v}) + \pi(\beta\vec{u}_2, \vec{v}) = \alpha\pi(\vec{u}_1, \vec{v}) + \beta\pi(\vec{u}_2, \vec{v})$
$2)\pi(\vec{u}, \alpha\vec{v}_1) + \pi(\vec{u}, \beta\vec{v}_2) = \alpha\pi(\vec{u}, \vec{v}_1) + \beta\pi(\vec{u}, \vec{v}_2)$

Definição VI.4.3 *Sejam $\mathcal{U}$ e $\mathcal{V}$ espaço vetoriais. A função $\phi : \mathcal{U} \times \mathcal{V} \longrightarrow \Re$ é um funcional bilinear se:*

1. $\phi(\alpha\vec{u}_1 + \beta\vec{u}_2, \vec{v}) = \alpha\phi(\vec{u}_1, \vec{v}) + \beta\phi(\vec{u}_2, \vec{v})$
2. $\phi(\vec{u}, \alpha\vec{v}_1 + \beta\vec{v}_2) = \alpha\phi(\vec{u}, \vec{v}_1) + \beta\phi(\vec{u}, \vec{v}_2)$

O espaço de todos funcionais bilineares assim definido é denotado por $\mathcal{U}^* \otimes \mathcal{V}^*$

Comentário:

$\mathcal{U}^* \otimes \mathcal{V}^*$ é também um espaço vetorial, já que as propriedades da adição e a multiplicação por escalar são herdadas pelos funcionais lineares.

VI.4.3 Produto Tensorial de Vetores

Definição VI.4.4 *Sejam $\mathcal{U}$ e $\mathcal{V}$ espaços vetoriais. O espaço produto tensorial $\mathcal{U} \otimes \mathcal{V}$ é $(\mathcal{U}^* \otimes \mathcal{V}^*)^*$. O produto tensorial de dois vetores $u \in \mathcal{U}$ e $v \in \mathcal{V}$ é :*

$$\vec{u} \otimes \vec{v} : \mathcal{U}^* \otimes \mathcal{V}^* \longrightarrow \Re; \phi \longmapsto \phi(\vec{u}, \vec{v})$$

Lema VI.4.3 $(\sum_i \alpha_i \vec{u}_i) \otimes (\sum_j \beta_j \vec{v}_j) = \sum_i \sum_j \alpha_i \beta_j \vec{u}_i \otimes \vec{v}_j$

Prova:

Usando as duas definições anteriores temos:

$$(\sum_i \alpha_i \vec{u}_i) \otimes (\sum_j \beta_j \vec{v}_j) : \phi \longmapsto \phi(\sum_i \alpha_i \vec{u}_i, \sum_j \beta \vec{v}_j) = \sum_i \sum_j \alpha\beta\phi(\vec{u}_i, \vec{v}_j) = (\sum_i \sum_j \alpha_i \beta_j \vec{u}_i \otimes \vec{v}_j)(\Phi)$$

Teorema VI.4.4 *Suponhamos que $\{\vec{u}_i\}$ é uma base de $\mathcal{U}$ e $\{\vec{v}_i\}$ é uma base de $\mathcal{V}$. Então $\{\vec{u}_i \otimes \vec{v}_j\}$ é uma base de $\mathcal{U} \otimes \mathcal{V}$. Assim se $\mathcal{U}$ e $\mathcal{V}$ têm dimensões n e m, então dimensão de $\mathcal{U} \otimes \mathcal{V}$ é igual a $m \times n$. Se os componentes de $\vec{u} \in \mathcal{U}$ com respeito à base $\{\vec{u}_i\}$ são $\{u_i\}$ e os componentes de $\vec{v} \in \mathcal{V}$ com respeito à base $\{\vec{v}_j\}$ são $\{v_j\}$, então os componentes de $\vec{u} \otimes \vec{v}$ com respeito à base $\{\vec{u}_i \otimes \vec{v}_j\}$ são $\{u_i v_j\}$.*

Prova:

Temos que $\{\hat{u}_i\}$ é base de $\mathcal{U}$ e $\{\hat{v}_j\}$ é base de $\mathcal{V}$. Vamos definir que:

$$\phi_{ij} = \phi(\vec{u}_i, \vec{v}_j)$$

Então,

$$\phi(\vec{u}, \vec{v}) = \phi(\sum_i u_i \vec{u}_i, \sum_j v_j \vec{v}_j)$$
$$\phi(\vec{u}, \vec{v}) = \sum_i \sum_j u_i v_j \phi(\vec{u}_i, \vec{v}_j)$$

$\phi(\vec{u},\vec{v}) = \sum_i \sum_j u_i v_j \phi_{ij}$

$\phi(\vec{u},\vec{v}) = \sum_i \sum_j \hat{u}_i^*(\vec{u})\hat{v}_j^*(\vec{v})\phi_{ij}$

$\phi(\vec{u},\vec{v}) = [\sum_i \sum_j \phi_{ij}\hat{u}_i^* \otimes \hat{v}_j^*](\vec{u},\vec{v})$

logo,

$$\phi = \sum_i \sum_j \phi_{ij}\hat{u}_i^* \otimes \hat{v}_j^*$$

Temos também que os funcionais $\{\hat{u}_i^* \otimes \hat{v}_j^*\}$ geram $\mathcal{U}^* \otimes \mathcal{V}^*$. A independência linear destes funcionais já foi demonstrada anteriormente logo, $\{\hat{u}_i^* \otimes \hat{v}_j^*\}$ é uma base para $\mathcal{U}^* \otimes \mathcal{V}^*$. Assim podemos utilizar sua base dual como uma base para $(\mathcal{U}^* \otimes \mathcal{V}^*)^* = \mathcal{U} \otimes \mathcal{V}$, pois existe um isomorfismo canônico. Seja essa base dual $\{f_{ij}\}$. Por lema anterior temos que f_{ij} é o funcional que extrai os componentes ϕ_{ij} de qualquer $\phi \in \mathcal{U}^* \otimes \mathcal{V}^*$ com respeito a uma base $\{\hat{u}_i^* \otimes \hat{v}_j^*\}$:

$$f_{ij} : \phi \longmapsto \phi_{ij} = \phi(\vec{u}_i,\vec{v}_j) = [\vec{u}_i \otimes \vec{v}_j](\phi)$$

isto é, $f_{ij} = \hat{u}_i \otimes \hat{v}_j$, e o conjunto $\{\hat{u}_i \otimes \hat{v}_j\}$ é exatamente à base de $\mathcal{U} \otimes \mathcal{V}$ dual a $\{\hat{u}_i^* \otimes \hat{v}_j^*\}$.

Resta verificar que se os componentes de $u \in \mathcal{U}$ com respeito à base $\{\hat{u}_i\}$ são $\{u_i\}$ e os componentes de $v \in \mathcal{V}$ com respeito à base $\{\hat{v}_j\}$ são $\{v_j\}$, então os componentes de $u \otimes v$ com respeito à base $\{\hat{u}_i \otimes \hat{v}_j\}$ são $\{u_i v_j\}$. Usando lema anterior temos:

$$\vec{u} \otimes \vec{v} = [\sum_i u_i \vec{\hat{u}}_i] \otimes [\sum_j v_j \vec{\hat{u}}_j] = \sum_i \sum_j (u_i v_j)(\vec{\hat{u}}_i \otimes \vec{\hat{v}}_j)$$

Assim, a componente ij de $\vec{u} \otimes \vec{v}$ com respeito à base $\{\vec{\hat{u}}_i \otimes \vec{\hat{v}}_j\}$ é simplesmente $u_i v_j$.

Teorema VI.4.5 *Se um conjunto $\{v_j\}$ é linearmente independente, e $\sum_j u_j \otimes v_j = 0$, então todos v_j são iguais 0.*

Prova:

Expandindo cada vetor u_j em uma base $\{u_i\}$ para $\mathcal{U}$, temos:

$$0 = \sum_j \vec{u}_j \otimes \vec{v}_j = \sum_j (\sum_i u_{j,i} \vec{\hat{u}}_j) \otimes \vec{v}_j = \sum_i \sum_j u_{j,i} \vec{\hat{u}}_i \otimes \vec{v}_j$$

Uma vez que $\{\vec{v}_i\}$ é linearmente independente, podemos adicionar outros vetores , se necessário, até que $\{\vec{v}_j\}$ se torne $\{\vec{\hat{v}}_j\}$, uma base para $\mathcal{V}$. Então a equação precedente dá uma expressão para o vetor nulo em $\mathcal{U} \otimes \mathcal{V}$ em termos da base $\{\vec{\hat{u}}_i \otimes \vec{\hat{v}}_j\}$. Como temos que o coeficiente do vetor da base deve sumir, logo, concluimos que todos $u_{i,j} = 0$, isto é, todos $\vec{u}_j = 0$.

Teorema VI.4.6 *Seja $\{\vec{\hat{v}}_j\}$ uma base de $\mathcal{V}$. Então todo vetor em $\mathcal{U} \otimes \mathcal{V}$ pode ser unicamente escrito: $\sum_j \vec{u}_j \otimes \vec{\hat{v}}_j$.*

Prova:

Uma vez que $\{\vec{\hat{u}}_i \otimes \vec{\hat{v}}_j\}$ é uma base para $\mathcal{U} \otimes \mathcal{V}$, todo vetor $\vec{w}$ pertencente a $\mathcal{U} \otimes \mathcal{V}$ pode ser escrito

$$\sum_i \sum_j w_{i,j} \vec{\hat{u}}_i \otimes \vec{\hat{v}}_j = \sum_j (\sum_i w_{i,j} \vec{\hat{u}}_i) \otimes \vec{\hat{v}}_j$$

Assim, podemos escolher $\vec{u}_i = \sum_j w_{i,j} \vec{\hat{u}}_i$ e teremos que $\vec{w} \in \mathcal{U} \otimes \mathcal{V}$ pode ser escrito como $\sum_j \vec{u}_j \otimes \vec{\hat{v}}_j$. A questão da unicidade é resolvida tendo como base teorema anterior, desta forma, se:

$$\sum_j \vec{u}_j \otimes \vec{v}_j = \sum_j \vec{u'}_j \otimes \vec{v}_j$$

então

$$\sum_j \vec{u}_j \otimes \vec{v}_j - \sum_j \vec{u'}_j \otimes \vec{v}_j = \sum_j (\vec{u}_j - \vec{u'}_j) \otimes \vec{v}_j = 0$$

daí e teorema anterior $\vec{u}_j - \vec{u'}_j = 0$, logo, $\vec{u}_j = \vec{u'}_j$

Teorema VI.4.7 $(\vec{u} \otimes \vec{v}).(\vec{u'} \otimes \vec{v'}) = (\vec{u}.\vec{u'}).(\vec{v}.\vec{v'})$

Prova:

Se $\mathcal{U}$ e $\mathcal{V}$ possuem produtos internos, existe um produto interno canônico sobre $\mathcal{U} \otimes \mathcal{V}$ em que a base $\{\vec{u}_i \otimes \vec{v}_j\}$ é ortonormal se $\{\vec{u}_i\}$ e $\{\vec{v}_j\}$ também são. Com este produto interno temos:

$$(\vec{u} \otimes \vec{v}).(\vec{u}' \otimes \vec{v}') = \sum_i \sum_j (\vec{u} \otimes \vec{v})_{ij}.(\vec{u}' \otimes \vec{v}')_{ij} = \sum_i \sum_j u_i v_j u'_i v'_j =$$
$$[\sum_i u_i u'_i][\sum_j u_j v'_j] = (\vec{u}.\vec{u}').(\vec{v}.\vec{v}')$$

Este resultado mostra que este produto interno sobre $\mathcal{U} \otimes \mathcal{V}$ é independente da escolha da base para $\mathcal{U}$ e $\mathcal{V}$, e depende somente dos seus produtos internos, isoladamente.

VI.5 Representação Conexionista por Produto Tensorial

Vejamos, agora, a formalização do conceito de representação conexionista e a formalização do problema da representação estruturada pela redução deste problema em três subproblemas:

- decomposição de estruturas simbólicas em papéis,
- representação de conjunções e
- representação das ligações variável/valor.

VI.5.1 Representação Conexionista

Definição VI.5.1 *Os estados de atividade de uma rede neuronal são os elementos de um espaço vetorial $\mathcal{V}$ que tem uma base distinguida $\{\hat{v}_i\}$.*

Comentário:

Um estado de atividade de uma rede neuronal é um vetor onde cada componente é o valor de ativação de uma unidade.Cada unidade corresponde a um vetor independente de uma base de um espaço vetorial cuja a dimensão é igual ao número de unidades da rede.Os valores de ativação das unidades não são restringidos para os estudos teóricos dos aspectos representacionais que estudaremos.

Definição VI.5.2 *Uma representação conexionista de uma estrutura simbólica em um conjunto $\mathcal{S}$ é uma função $\Psi : \mathcal{S} \longrightarrow \mathcal{V}$, onde $\mathcal{V}$ é um espaço vetorial.*

Comentário:

O conjunto $\mathcal{S}$ de estruturas simbólicas aqui estudadas estão restringidas a conjuntos de cadeia de caracteres e conjuntos de árvores binárias.

Definição VI.5.3 *Uma representação conexionista Ψ é* precisa *("faithful") se e somente se, ela é uma bijeção e não existe nenhuma estrutura que é mapeada no vetor nulo $O \in \mathcal{V}$.*

Comentário:

Está definição estabelece as condições em que estruturas simbólicas distintas possuem representações distintas.

VI.5.2 Decomposição de Estruturas Simbólicas

Como veremos ao longo deste capítulo, uma representação por produto tensorial de um conjunto de estruturas S atribui para cada $s \in S$ um vetor construído pela superposição das representações de seus constituintes, isto é, cada objeto $s \in S$, onde S é um conjunto de estruturas simbólicas, é representado pela superposição das representações dos constituintes deste objeto, onde os constituintes deste objeto estruturado são especificados por uma certa decomposição em papéis.

A representação de objetos estruturados tratada por Smolensky é do tipo em que os objetos podem ser vistos como possuidores de um número (possivelmente ilimitado) de papéis que, para instâncias particulares da estrutura, são ligados individualmente a preenchedores particulares.Assim, uma decomposição em papéis de S especifica um constituinte de $s \in S$ pela atribuição a ele de um conjunto não ordenado de ligações preenchedor/papel. Vejamos um exemplo.Seja S o conjunto de *"strings"* formadas a partir do alfabeto $\{a, b, c\}$, e $s = cba$, então poderíamos escolher uma decomposição em papéis em que os papéis são as posições absolutas dos

caracteres dentro da *"string"* e os constituintes são as ligações preenchedor/papel, isto é, $\{b/r_2, a/r_3, c/r_1\}$ .De forma mais generalizada teríamos que uma *"string"* pode ser vista como possuidora de um conjunto de papéis $\{r_1, r_2, \cdots\}$ onde r_i é o papel do i-ésimo elemento da *"string"*.

Outros tipos de decomposições em papéis também são possíveis, por exemplo, poderíamos ter uma decomposição em papéis, que ao invés de considerar a posição absoluta dentro de uma *"string"*, considera um contexto local, isto é, em vez de r_i como o i-ésimo elemento de uma *"string"* teríamos r_{x_y} (é precedido por x e seguido de y, para certos valores de x e y).

Um outro ponto interessante é que quando os preenchedores ou papéis são também objetos estruturados podemos aplicar outras decomposições em papéis a fim de reduzí-los ainda mais. Caso os preenchedores ou papéis sejam estruturas do mesmo tipo do objeto estruturado original podemos aplicar a mesma decomposição em papéis, isto é, podemos utilizar a decomposição em papéis de forma recursiva.

Definição VI.5.4 *Seja S um conjunto de estruturas simbólicas. Uma "decomposição em papéis"("role decomposition") F/R para S é um par de conjuntos (F, R), onde F é um conjunto de preenchedores e R é um conjunto de papéis, e uma função $\mu_{F/R} : F \times R \longrightarrow Pred(S)$ e para qualquer par $f \in F, r \in R$, o predicado sobre S:*

$$\mu_{F/R}(f, r) = f/r$$

é expresso assim: f preenche o papel ("role") r.

Comentário:

A decomposição em papéis, como vimos anteriormente, é um método para a redução de complexos itens estruturados em conjuntos de pares. Por exemplo, o conceito complexo *"garrafa de vinho"* pode ser analisado como tendo dois papéis : *"contém"* e *"contido por"*. O primeiro papel é ocupado por *"garrafa"* e o segundo por *"vinho"*.

Definição VI.5.5 *A decomposição em papéis possui papéis de valores únicos ("single-valued roles") se, e somente se, para qualquer $s \in S$ e $r \in R$, existe no máximo um $f \in F$ tal que $f/r(s)$ valha.*

Definição VI.5.6 *Uma decomposição em papéis é recursiva se, e somente se, $F = S$.*

Definição VI.5.7 *Uma decomposição em papéis determina um mapeamento :*

$$\beta : S \longrightarrow 2^{F \times R}; s \longmapsto \{(f, r) \mid f/r(s)\}$$

que associa a cada $s \in S$ o conjunto $\beta(s)$ de ligações preenchedor/papel em S. O mapeamento β será chamado de representação preenchedor/papel de S induzida pela decomposição em papéis.

Definição VI.5.8 *Uma decomposição em papéis é precisa se, e somente se, β é um mapeamento 1-1.*

Comentário:

As decomposição em papéis precisas são particularmente úteis pois, as representações preenchedor/papel por elas induzidas, nos permitem identificar uma estrutura simbólica com um predicado tendo uma forma conjuntiva simples, que é a base da representação por produto tensorial.

Definição VI.5.9 *Uma decomposição é finita se, e somente se, para cada $s \in S$, o conjunto de ligações em s, $\beta(s)$, é finito.*

Teorema VI.5.1 *Seja F/R uma decomposição em papéis para S. Para cada $s_0 \in S$, defini-se um predicado π_{s_0} por:*

$$\pi_{s_0}(s) = \bigwedge_{(f,r) \in \beta(s_0)} f/r(s)$$

onde $\wedge$ denota conjunção; $\pi_{s_0}(s)$ é verdadeiro se, e somente se, a estrutura s contém todas as ligações f/r em s_0. Assim, se a decomposição em papéis é precisa, a estrutura s_0 pode ser recuperada do predicado π_{s_0}

Prova:

Segue imediatamente a partir do próximo lema.

Lema VI.5.1 *A função β de uma decomposição em papéis mapeia elementos de S em subconjuntos de $F \times R$. Estes subconjuntos possuem uma ordem parcial $\subseteq$ que pode ser retirada de S via β:*

$$s_1 \leq s_2 sss \beta(s_1) \subseteq \beta(s_2).$$

Suponha que F/R é precisa. Então, com respeito a ordem parcial $\leq$, o conjunto de elementos de S para o qual o predicado π_{s_0} vale tem um único elemento ínfimo que é s_0. Desta maneira s_0 pode ser recuperado do predicado π_{s_0}.

Prova:

Temos que $\beta(s)$ é o conjunto de ligações preenchedor/papel em s, $s_1 \leq s_2$ se, e somente se, as ligações em s_1 formam subconjuntos das ligações s_2:

$$s_1 \leq s_2 \text{ sss } [\text{ para todo } f \in F \text{ e } r \in R, f/r(s_1) \Rightarrow f/r(s_2)].$$

Considerando o conjunto de elementos s satisfazendo o predicado π_{s_0}, temos:
$S(\pi_{s_0}) = \{s \in S \mid \pi_{s_0}\}$
$S(\pi_{s_0}) = \{s \in S \mid \text{para todo } f \in F \text{ e } r \in R, f/r(s_0) \Rightarrow f/r(s)\}$
$S(\pi_{s_0}) = \{s \in S \mid s_0 \leq s\}$.

Este conjunto ($S(\pi_{s_0})$) contém s_0, e s_0 é um ínfimo. Temos que mostrar que este ínfimo é único. Assim, consideremos um outro elemento $s_1 \in S(\pi_{s_0})$. Uma vez que μ é precisa e $s_1 \neq s_0$, e existe no mínimo uma ligação f/r não compartilhada por s_0 e s_1. Uma vez que $s_1 \in S(\pi_{s_0})$ e s_0 é um ínfimo de $S(\pi_{s_0})$, devemos ter $f/r(s_1) \wedge \neg f/r(s_0)$. Daí, $\neg(s_1 \neq s_0)$, que é um absurdo, logo s_1 não pode se um ínfimo de $S(\pi_{s_0})$, isto é, s_0 é ínfimo e único.

VI.5.3 Representação Conexionista de Conjunções

Definição VI.5.10 *Uma representação conexionista Ψ emprega a representação por superposição de conjunção se, e somente se:*

$$\Psi(\Lambda_i p_i) = \sum_i \Psi(p_i).$$

Comentário:

Na realidade, a representação de conjunções é uma soma vetorial, isto é, se duas proposições são representadas cada uma por uma por um padrão de atividade, a conjunção destas proposições é o padrão de atividade resultante da superposição dos padrões individuais (soma). Esta definição estabelece, assim, que a representação da conjunção de uma coleção de proposições é a soma das representações individuais de cada proposição.

Definição VI.5.11 *Seja S um conjunto de estruturas simbólicas e F/R uma decomposição em papéis de S . Suponhamos que Ψ_b é uma representação das ligações f/r:*

$$\Psi_b : \{f/r \mid f \in F, r \in R\} \longrightarrow \mathcal{V}$$

onde $\mathcal{V}$ é um espaço vetorial. Então $\Psi_{F/R}$, a representação conexionista induzida por F/R, a representação superposicional de conjunção, e Ψ_b é:

$$\Psi_{F/R} : S \longrightarrow \mathcal{V}; s \longmapsto \sum_{(f,r)\in\beta(s)} \Psi_b(f/r).$$

Um exemplo

Analisemos, agora, este exemplo simples a fim de tornar mais claro alguns conceitos e definições da decomposição em papéis de uma estrutura simbólica S.

Seja S o seguinte conjunto de "strings" formado a partir do alfabeto $\{a, b, c\}$: $S = \{ab, bc, abc, cba\}$. Desta forma, considerando a decomposição em

papéis como a decomposição de uma estrutura $s \in S$ em ligações preenchedor/papel, tendo como referência as posições absolutas ocupadas pelos símbolos, teríamos:

- conjunto de papéis: $R = \{r_1, r_2, r_3\}$
- conjunto de preenchedores: $F = \{a, b, c\}$
- $F \times R = \{(a, r_1), (a, r_2), (a, r_3), (b, r_1), (b, r_2), (b, r_3), (c, r_1), (c, r_2), (c, r_3)\}$

Os valores da função $\mu_{F/R} : F \times R \longrightarrow Pred(S)$ são:

- $\mu_{F/R}(a, r_1) = a/r_1$
- $\mu_{F/R}(b, r_2) = b/r_2$
- $\mu_{F/R}(b, r_1) = b/r_1$
- $\mu_{F/R}(c, r_2) = c/r_2$
- $\mu_{F/R}(a, r_1) = a/r_1$
- $\mu_{F/R}(b, r_2) = b/r_2$
- $\mu_{F/R}(c, r_3) = c/r_3$
- $\mu_{F/R}(c, r_1) = c/r_1$
- $\mu_{F/R}(b, r_2) = b/r_2$
- $\mu_{F/R}(a, r_3) = a/r_3$

Assim, os conjuntos $\beta(s)$ são:

- $\beta(ab) = \{a/r_1, b/r_2\}$
- $\beta(bc) = \{b/r_1, c/r_2\}$
- $\beta(abc) = \{a/r_1, b/r_2, c/r_3\}$
- $\beta(cba) = \{c/r_1, b/r_2, a/r_3\}$

e, os predicados $\pi_{s_0}(s)$:

- $\pi_{ab}(s) = a/r_1(s) \wedge b/r_2(s)$
- $\pi_{bc}(s) = b/r_1(s) \wedge c/r_2(s)$
- $\pi_{abc}(s) = a/r_1(s) \wedge b/r_2(s) \wedge c/r_3(s)$
- $\pi_{cba}(s) = c/r_1(s) \wedge b/r_2(s) \wedge a/r_3(s)$

VI.5.4 Representação Conexionista da Ligação de Variáveis

Veremos, agora, como representar a ligação de variáveis. Para ligar um preenchedor f a um papel r, temos que primeiro ter f como um padrão de atividade $\vec{f}$ sobre um conjunto de unidades-preenchedor $\{\tilde{f}_\phi\}$ e r como um padrão de atividade $\vec{r}$ sobre um conjunto de unidade-papel $\{\tilde{r}_\rho\}$. A ligação f/r é representada pelo padrão de atividade $f\overrightarrow{/}r$ sobre um conjunto de unidade-ligação $\{\tilde{b}_{\phi\rho}\}$. O valor de atividade da unidade $\tilde{b}_{\phi\rho}$ é a atividade $\tilde{f}_\phi$ do padrão f vezes a atividade da unidade $\tilde{r}_\rho$ do padrão r.

Na terminologia da teoria dos espaços vetoriais, teríamos:

- a representação de um papel r é um vetor $\vec{r}$ em um espaço vetorial $\mathcal{V}_R$, que tem dimensão igual ao número de unidades-papel $\tilde{r}_\rho$.
- a representação de um preenchedor f é um vetor $\vec{f}$ em um espaço vetorial $\mathcal{V}_F$, que tem dimensão igual ao número de unidades-preenchedor $\tilde{f}_\phi$.
- a representação da ligação f/r é um vetor produto tensorial $f\overrightarrow{/}r = f \otimes r$ no espaço vetorial real $\mathcal{V}_B = \mathcal{V}_F \otimes \mathcal{V}_R$, e $\mathcal{V}_B$ tem a dimensão é igual ao produto da dimensão de $\mathcal{V}_F$ vezes a dimensão de $\mathcal{V}_R$.

Definição VI.5.12 *Seja F/R uma decomposição em papel de S. Sejam Ψ_F e Ψ_R representações conexionistas de preenchedores e papéis:*

$$\Psi_F : F \longrightarrow \mathcal{V}_F$$
$$\Psi_R : R \longrightarrow \mathcal{V}_R$$

Então a Representação por Produto Tensorial das ligações preenchedor/papel induzida por Ψ_F e Ψ_R é o mapeamento:

$$\Psi_b : \{f/r \mid f \in F, r \in R\} \longrightarrow \mathcal{V}_F \otimes \mathcal{V}_R; f/r \longrightarrow \Psi_F(f) \otimes \Psi_R(r)$$

VI.5.5 Representação por Produto Tensorial

Vamos definir, agora, a representação por produto tensorial de uma estrutura S a partir das seguintes representações já definidas:

- representação conexionista de conjunções
- representação conexionista de preenchedores
- representação conexionista de papéis
- representação conexionista de ligações preenchedor/papel

Definição VI.5.13 *Seja F/R uma decomposição em papel de S e sejam Ψ_F e Ψ_R representações conexionistas de preenchedores e papéis.Então a Representação por Produto Tensorial correspondente de S é:*

$$\Psi : S \longrightarrow \mathcal{V}_F \otimes \mathcal{V}_R; s \longmapsto \sum_{(f,r)\in\beta(s)} \Psi_F(f) \otimes \Psi_R(r)$$

Então a Representação por Produto Tensorial das ligações preenchedor/papel induzida por Ψ_F e Ψ_R é o mapeamento:

$$\Psi_b : \{f/r \mid f \in F, r \in R\} \longrightarrow \mathcal{V}_F \otimes \mathcal{V}_R; f/r \longrightarrow \Psi_F(f) \otimes \Psi_R(r)$$

Se identificarmos s com as conjunções das ligações e, considerarmos $\vec{f} = \Psi_F(f)$ e $\vec{r} = \Psi_R(r)$, teremos:

$$\Psi(\wedge_i f_i/r_i) = \sum_i \vec{f_i} \otimes \vec{r_i}.$$

VI.6 Tipos de Representações da RTP

Definição VI.6.1 *Seja Ψ uma representação conexionista de um conjunto $\mathcal{X}$ em um espaço vetorial $\mathcal{V}$ com uma base distinguida $\{\vec{v}_i\}$. Ψ é uma representação local se, e somente se, é um mapeamento 1-1 de elementos de $\mathcal{X}$ sobre o conjunto de vetores da base $\{\vec{v}_i\}$. Se uma representação conexionista não é local, então é distribuída.*

Definição VI.6.2 *Seja $\Psi_{F/R}$ uma representação por produto tensorial de S induzida por uma decomposição em papéis F/R de S e, Ψ_F e Ψ_R representações conexionistas. Então $\Psi_{F/R}$ é uma representação por produto tensorial puramente local se Ψ_F e Ψ_R são ambas representações locais.*

Definição VI.6.3 *Seja $\Psi_{F/R}$ uma representação por produto tensorial de S induzida por uma decomposição em papéis F/R de S e, Ψ_F e Ψ_R representações conexionistas. Se Ψ_F é uma representação distribuída e Ψ_R é uma representação local, então $\Psi_{F/R}$ é uma representação por produto tensorial parcialmente local.*

Definição VI.6.4 *Seja $\Psi_{F/R}$ uma representação por produto tensorial de S induzida por uma decomposição em papéis F/R de S e, Ψ_F e Ψ_R representações conexionistas. Se Ψ_F e Ψ_R são ambas representações distribuídas, então $\Psi_{F/R}$ é uma representação por produto tensorial completamente distribuída.*

VI.7 "Unbinding"

Teorema VI.7.1 *Seja $\Psi_{F/R}$ uma representação por produto tensorial induzida por uma decomposição em papéis com papéis de valores simples. Suponhamos que os vetores representante dos papéis ligados em uma estrutura s são todos linearmente independente. Então cada papel pode se tornar livre ("unbound") com completa precisão, isto é, para cada papel ligado r_i existe uma operação que toma o vetor $\Psi_{F/R}(s)$ representando s e leva ao vetor $\vec{f}_i$ representando o preenchedor f_i ligado a r_i.*

Prova:

Se os vetores representativos dos papéis $\{\vec{r}_i\}$ são linearmente independente, então eles formam uma base para o subespaço de $\mathcal{V}_R$ por eles gerado. Para esta base temos uma base dual $\{U_i\}$ correspondente. Cada elemento nesta base dual é um mapeamento linear de $\mathcal{V}_R$ no números reais com a seguinte propriedade:

$$U_i(r_j) = \delta_{ij} = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$$

Assim, temos que: U_i mapeia o vetor $\vec{r}_i$ em 1 e todos os outros em 0. Se nós utilizarmos o produto interno canônico no espaço vetorial $\mathcal{V}_R$ então, o vetor dual U_i pode ser expresso como uma operação que toma o produto interno com respeito a algum vetor $\vec{u}_i$ em $\mathcal{V}_R$, isto é,

$$U_i(\vec{v}) = \vec{v}\vec{u}_i$$

para todo $\vec{v}$ em $\mathcal{V}_R$. Chamemos $\{\vec{u}_i\}$ de vetores livres (*"unbinding vectors"*) para papéis $\{r_i\}$. Tomemos $\vec{s}$ como sendo a representação por produto tensorial de uma estrutura em que os papéis $\{\vec{r}_i\}$ estão ligados aos preenchedores $\{f_i\}$. Então nós podemos extrair f_i de $\vec{s}$, ou tornar r_i livre, tomando o produto interno parcial de $\vec{s}$ com o vetor livre $\vec{u}_i$:

$$\vec{s}\vec{u}_i = (\sum_j \vec{f}_j \otimes \vec{r}_j)\vec{u}_i = \sum_j \vec{f}_j(\vec{r}_j\vec{r}_i) = \sum_j \vec{f}_j\delta_{ij} = \vec{f}_i$$

Definição VI.7.1 *O procedimento definido na prova anterior é o procedimento de liberação exata (*"exact unbinding procedure"*). Suponha que a liberação do papel $\vec{r}_i$ é realizado como na prova anterior, mas substituímos o vetor livre $\vec{u}_i$ pelo próprio vetor $\vec{r}_i$. Temos, desse modo, o procedimento de liberação auto-endereçavél (*self-addressing unbinding procedure*).*

Comentário:

Ao contrário do procedimento de liberação exata, o procedimento de liberação auto-endereçavél é definida para um conjunto qualquer de vetores representativos de papéis, até mesmo se eles não sejam linearmente independente.

Teorema VI.7.2 *Suponhamos que estamos utilizando o procedimento de liberação auto-endereçavél para liberação de papéis. Se os vetores representativos dos papéis são todos ortogonais, o padrão preenchedor correto será gerado (isto é, o vetor representativo do preenchedor será recuperado), desassociado, entretanto, do fator de magnitude como um todo (isto é, o vetor representativo do preenchedor não terá o mesmo módulo do vetor original). No outro caso (os vetores representativos dos papéis não são todos ortogonais), o padrão gerado será uma superposição ponderada do padrão correto do preenchedor $\vec{f}_i$ e todos os outros preenchedores $\{\vec{f}_j\}_{j\neq i}$. Nesta superposição, o peso de cada padrão errado $\vec{f}_j$ relativo ao padrão correto $\vec{f}_i$, isto é, a interferência do papel j no papel i é:*

$$\frac{\vec{r}_i\vec{r}_j}{\|\vec{r}_i\|^2} = cos\Theta ji\frac{\|\vec{r}_j\|}{\|\vec{r}_i\|}$$

onde Θji é o ângulo entre os vetores $\vec{r}_j$ e $\vec{r}_i$

Prova:

Considerando a utilização do procedimento de liberação auto-endereçavél para a liberação de papéis temos:

$$\vec{s}\vec{r}_i = (\sum_j \vec{f}_j \oslash \vec{r}_j)\vec{r}_i = \sum_j \vec{f}_j(\vec{r}_j\vec{r}_i) = (\vec{r}_i\vec{r}_i)\vec{f}_i + \sum_{j\neq i}(\vec{r}_j\vec{r}_i)\vec{f}_j$$

Assim, nesta superposição ponderada, o raio do coeficiente de cada preechedor incorreto f_j para o preenchedor correto $\vec{f}_i$ é:

$$\frac{\vec{r}_i\vec{r}_j}{\vec{r}_i\vec{r}_i}$$

Já que o denominador é $\| \vec{r}_i \|^2$ e o numerador é $cos\Theta ji \| \vec{r}_j \| \| \vec{r}_i \|$, assim, temos a resultado desejado.

Para uma análise das outras propriedades da representação tensorial veja [SMOL87] e [SMOL90].

VI.8 Estudo de um Exemplo: "Frames"

Estudaremos, agora, um exemplo de aplicação da Representação por Produto Tensorial em representação estruturada do conhecimento. Mais especificamente, aplicaremos a *RTP* em um exemplo simples de *"frames"* utilizado em [DOLAN87] e [DOLAN88] e citado em por Smolensky em [SMOL89]. Porém, em nenhuma referência anterior foi apresentado o exemplo de forma detalhada. Desta forma, desenvolvemos, com relativo detalhamento, uma memória conexionista capaz de armazenar um conjunto de *"frames"* e instâncias desses *"frames"* e sendo capaz de realizar ligação de variáveis e o *"unbinding"* das mesmas.

VI.8.1 O modelo básico

Como dito em [SHAR91], uma maneira de se pensar os símbolos, em modelos conexionistas distribuído, é como uma cadeia de bits. Esta cadeia de bits é na realidade (para os modelos conexionistas) um vetor de características ao invés de endereços de memórias como nos sistemas simbólicos tradicionais. Desta forma, um número de bits é alocado para cada dimensão, e no nosso caso um bit para cada possível característica.

Considerando inicialmente que temos duas dimensões: *gênero* e *estatura*, podemos associar quatro característica, duas para cada dimensão. A dimensão gênero será formada pelas características *masculino* e *feminino* enquanto a dimensão *estatura* é formada pelas características *média* e *grande*. O mapeamento simbólico assumido no nosso exemplo é o seguinte (veja a tabela VI.1):

Nome → (gênero estatura)

Podemos, assim, interpretar que João é do gênero masculino e de estatura grande e Maria é do genero feminino e de estatura média. Consideremos agora que desejamos especificar os vetores representativos para um *"frame"* e dois de seus *"slots"*. Seguindo a descrição em [DOLAN88] reutilizaremos os mesmos

Símbolos	Vetores
João	(10 01)
Maria	(01 10)

Tabela VI.1: Tabela de símbolos e vetores característica

padrões de bits nesta exposição.Assim temos:

$$\text{FERIR} \rightarrow (1010)$$
$$\text{sujeito} \rightarrow (1010)$$
$$\text{objeto} \rightarrow (0101)$$

A partir da definição dos símbolos acima podemos construir uma relação entre símbolos pela concatenação dos vetores de bits representativos dos símbolos. Exemplificando, podemos representar *"João feriu Maria"* através de duas relações (101010101001) e (101001010110), (FERIR sujeito João) e (FERIR objeto Maria), respectivamente. Na terminologia tradicional da representação do conhecimento por *"frames"* teríamos, desta forma, *(FRAME SLOT FILLER)* (veja tabela VI.2).

FRAME	SLOT	FILLER
FERIR	sujeito	João
FERIR	objeto	Maria

Tabela VI.2: Representação do conceito "João feriu Maria"

VI.8.2 A RTP do "FRAME" Ferir

Utilizaremos os vetores definidos anteriormente para os seguintes símbolos: João, Maria, Ferir, sujeito, objeto (veja tabela VI.3).

Teremos que calcular os seguintes produtos tensoriais:

- Ferir $\otimes$ sujeito $\otimes$ João

Símbolo	Vetor
João	$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$
Maria	$\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$
Feriu	$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$
sujeito	$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$
objeto	$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$

Tabela VI.3: Vetores associados ao conceito "João feriu Maria"

- Ferir $\otimes$ objeto $\otimes$ Maria

Os cálculos são realizados considerando-se que o produto tensorial pode ser entendido como uma generalização do produto externo de dois vetores. Assim, dados dois vetores coluna x e y, o produto interno $x.y$ é $x^t.y$ e o produto externo é $x.y^t$. Como estaremos entendo o produto tensorial como um produto externo, teremos:

$$x \otimes y = x.y^t$$

isto é, o produto tensorial de dois vetores (tensores de "rank" 1) é uma matriz (tensor de "rank" 2). Para calcularmos o produto interno de 3 vetores ($x \otimes y \otimes z$), primeiro calculamos $x \otimes y$ e depois multiplicamos escalarmente esta matriz por cada componente de z (os escalares z_i). Veja [SMOL88] e [SMOL89] para maiores detalhes.

Cálculo de $FERIR \otimes sujeito \oslash Joao$

Temos que:

A) $FERIR \oslash sujeito = FERIR.(sujeito)^t =$

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} . \begin{pmatrix} 1 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

B) $(FERIR \otimes sujeito) \otimes Joao = (FERIR \otimes sujeito).Joao_k, k = 1 \cdots 4.$

- $(FERIR \otimes sujeito).Joao_1$

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} .1 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- $(FERIR \otimes sujeito).Joao_2$

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} .0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- $(FERIR \otimes sujeito).Joao_3$

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} .0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- $(FERIR \otimes sujeito).Joao_4$

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} .1 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Logo, o "Frame" (FERIR Sujeito João) é representado pelo tensor (vetor):

$$\begin{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{pmatrix}$$

Cálculo de $FERIR \otimes objeto \otimes Maria$

Temos que:

A) $FERIR \otimes objeto = FERIR.(objeto)^t =$

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} . \begin{pmatrix} 0 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

B) $(FERIR \otimes objeto) \otimes Maria = (FERIR \otimes objeto).Maria_k, k = 1 \cdots 4.$

- $(FERIR \otimes objeto).Maria_1$

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} .0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- $(FERIR \otimes objeto).Maria_2$

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} .1 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- $(FERIR \otimes objeto).Maria_3$

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} .1 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- $(FERIR \otimes objeto).Maria_4$

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} .0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Logo, o "Frame" (FERIR Objeto Maria) é representado pelo tensor (vetor):

$$\begin{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{pmatrix}$$

Vejamos como, a partir do tensor representante de uma instância, podemos realizar o "unbinding" do preechedor de um "slot" de um "frame". A propriedade utilizada é a seguinte :

$$(a \otimes b \otimes c).(d \otimes e) = (a.d).(b.e).c$$

No nosso caso:

$$(frame \otimes slot \otimes preenchedor).(frame \otimes slot) = \alpha.(preenchedor)$$

onde,

$$\alpha = (frame.frame)(slot.slot)$$

No caso do frame (FERIR Sujeito João):

$$\left(\left(\begin{pmatrix}1\\0\\1\\0\end{pmatrix}\otimes\begin{pmatrix}1\\0\\1\\0\end{pmatrix}\otimes\begin{pmatrix}1\\0\\0\\1\end{pmatrix}\right)\cdot\left(\left(\begin{pmatrix}1\\0\\1\\0\end{pmatrix}\otimes\begin{pmatrix}1\\0\\1\\0\end{pmatrix}\right)\right)=$$

$$\left(\begin{pmatrix}\begin{pmatrix}1&0&1&0\\0&0&0&0\\1&0&1&0\\0&0&0&0\end{pmatrix}\\\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\\\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\\\begin{pmatrix}1&0&1&0\\0&0&0&0\\1&0&1&0\\0&0&0&0\end{pmatrix}\end{pmatrix}\cdot\begin{pmatrix}1&0&1&0\\0&0&0&0\\1&0&1&0\\0&0&0&0\end{pmatrix}\right)=$$

$$\begin{pmatrix}1&0&1&0\\0&0&0&0\\0&0&0&0\\1&0&1&0\end{pmatrix}\cdot\begin{pmatrix}1\\0\\1\\0\end{pmatrix}=\begin{pmatrix}1\\0\\0\\1\end{pmatrix}.1+\begin{pmatrix}0\\0\\0\\0\end{pmatrix}.0+\begin{pmatrix}1\\0\\0\\1\end{pmatrix}.1+\begin{pmatrix}0\\0\\0\\0\end{pmatrix}.0=\begin{pmatrix}2\\0\\0\\2\end{pmatrix}$$

$$\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\cdot\begin{pmatrix}0\\0\\0\\0\end{pmatrix}=\begin{pmatrix}0\\0\\0\\0\end{pmatrix}.0+\begin{pmatrix}0\\0\\0\\0\end{pmatrix}.0+\begin{pmatrix}0\\0\\0\\0\end{pmatrix}.0+\begin{pmatrix}0\\0\\0\\0\end{pmatrix}.0=\begin{pmatrix}0\\0\\0\\0\end{pmatrix}$$

$$\begin{pmatrix}1&0&1&0\\0&0&0&0\\0&0&0&0\\1&0&1&0\end{pmatrix}\cdot\begin{pmatrix}1\\0\\1\\0\end{pmatrix}=\begin{pmatrix}1\\0\\0\\1\end{pmatrix}.1+\begin{pmatrix}0\\0\\0\\0\end{pmatrix}.0+\begin{pmatrix}1\\0\\0\\1\end{pmatrix}.1+\begin{pmatrix}0\\0\\0\\0\end{pmatrix}.0=\begin{pmatrix}2\\0\\0\\2\end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Assim,

$$\begin{pmatrix} 2 \\ 0 \\ 0 \\ 2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 2 \\ 0 \\ 0 \\ 2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \\ 0 \\ 4 \end{pmatrix} = 4. \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

No caso do frame (FERIR Objeto Maria):

$$\left(\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \right) \cdot \left(\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \right) =$$

$$\begin{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} =$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} .1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .1 = \begin{pmatrix} 0 \\ 2 \\ 2 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} .1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .1 = \begin{pmatrix} 0 \\ 2 \\ 2 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Assim,

$$\begin{pmatrix} 0 \\ 2 \\ 2 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 2 \\ 2 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \\ 4 \\ 0 \end{pmatrix} = 4. \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

Mostramos, assim, a representação dos "frames"

- Ferir ⊗ sujeito ⊘ João
- Ferir ⊗ objeto ⊗ Maria

podemos representar toda a estrutura pela equação:

$$\sum_i s_i \oslash r_i \oslash f_i$$

no nosso caso, teríamos:

$$\sum_i frame_i \otimes slot_i \otimes preenchedor_i$$

Considerando $i = 1$ para João e $i = 2$ para Maria temos que a estrutura será representada pelo seguinte tensor:

(Ferir ⊗ sujeito ⊗ João) + (Ferir ⊗ objeto ⊗ Maria)

isto é,

$$\left(\begin{pmatrix}\begin{pmatrix}1&0&1&0\\0&0&0&0\\1&0&1&0\\0&0&0&0\end{pmatrix}&\begin{pmatrix}0&1&0&1\\0&0&0&0\\0&1&0&1\\0&0&0&0\end{pmatrix}\\\begin{pmatrix}0&1&0&1\\0&0&0&0\\0&1&0&1\\0&0&0&0\end{pmatrix}&\begin{pmatrix}1&0&1&0\\0&0&0&0\\1&0&1&0\\0&0&0&0\end{pmatrix}\end{pmatrix}\right)$$

Para uma análise de como este procedimento aqui explicitado serviu de base para a arquitetura de um sistema que trabalha com manipulação de conhecimento conceitual veja [DOLAN87] e [DOLAN88].

Dois pontos que devem ser ressaltados é que o sistema DCPS estudado no capítulo V foi completamente re-implementado usando o formalismo do produto tensorial. Este novo sistema recebeu o nome de sistema de produção por produto tensorial - TPPS - (Tensor Product Production System) e mostra a viabilidade desta teoria, para maiores detalhes veja [SMOL89]. O outro ponto é que o representação por produto tensorial pode ser ampliada para a manipulação de estruturas recursivas que são de fundamental importância para o PCLN, veja [LEGE91].

Capítulo VII

Conclusões

Neste trabalho, procuramos reunir os diversos conceitos, já estabelecidos, respeito da representação do conhecimento em modelos conexionista, isto é, os tipos de representação, suas características e propriedades. Mostramos, também, a necessidade de futuros trabalhos que busquem uma combinação das diversas características positivas de cada forma de representação .

Acreditamos, que a integração entre as diversas metáforas para a cognição humana (e para inteligência artificial) pode estabelecer novas formulações (e soluções) de antigos problemas. Especialmente, a combinação de técnicas conexionista para a representação e manipulação de estruturas simbólicas.

Constatamos, que a pesquisa da representação do conhecimento simbólico em modelos conexionista ainda está na fase inicial, e portanto, não apresenta, ainda, posições claras compartilhadas por todos pesquisadores.

Quanto a teoria da representação por produto tensorial, desenvolvida pelo Prof. Paul Smolensky, acreditamos que poderá ser aplicada em diversos modelos e extrapolada para outros domínios que necessitem de uma abordagem conjunta de seus aspectos discreto e contínuo. Cabe ainda dizer que apresentamos apenas os conceitos básicos da representação por produto tensorial e que uma gama imensa de tópicos se apresenta para um estudo bem mais detalhado. Temos, por exemplo, que a aplicação da recirculação simbólica no aprendizado das representações merece ser detalhada e explorada nas suas aplicações à estruturas simbólicas, especialmente

quando aplicadas ao processamento conexionista da linguagem natural.

Uma aplicação recente da representação por produto tensorial pode ser encontrada em [LEGE91] e [SMOL92], onde é desenvolvido uma teoria conexionista para o PCLN baseada na gramática harmônica (veja [LEGE90]) Esta teoria combina dois níveis de abordagem: um utilizando a representação distribuída e o outro a local, ambos sob o formalismo do produto tensorial, para um tratamento de certos problemas sintáticos do processamento da linguagem natural. Isto confirma nossa visão que devemos, quase sempre, combinar os diversos tipos de representação tendando aproveitar as vantagens e amenizar as características indesejáveis inerentes a cada tipo de representação isoladamente.

Referências Bibliográficas

[ANDER77] ANDERSON, J.A., SILVERSTEIN, J.W., RITZ, S.A. e JONES, R.S. , "Distinctive Features, Categorical Perception, and Probability Learning, *Psychological Review*, No.84, 1977, 413-451.

[ARARI88] ARARIBÓIA,G., *Inteligência Artificial*, Livros Técnicos e Científicos,1988.

[BOBR75] BOBROW, D.G. e NORMAN, D.A., *Some Principles of Memory Schemata*. In D.G. BobrowMir & Collins (eds.), Representation and Understanding, 131-149. NY: Academic Press , 1975.

[BUCH84] BUCHANAN,B.G. e SHORTIFFE,E.H. (eds.),*Rule-Based Expert Systems - The Mycin Experiments of the Stanford Heuristic Programing Project*, Addison-Wesley Publishing,1984.

[CHAN88] CHANDRASEKARAN, B.,"Connectionism and Information Processing Abstractions", *AI Magazine*, Vol 9, No.4, 1988, 24-34.

[CARV89a] CARVALHO,L.A.V., *Síntese de Redes Neuronais com Aplicações à Representação do Conhecimento e à Otimização*, Tese de Doutorado, COPPE - Sistemas, UFRJ, 1989.

[CARV89b] CARVALHO,L.A.V., *Redes Neuronais - Notas de Aula*, COPPE - Sistemas, UFRJ, 1989.

[ACAR91] CARVALHO A. A., "Objetivos e Desenvolvimento da Quinta Geração de Computadores", *Relatório Técnico-Científico, COPPE-PADCT-CNPq*, 1991,87-124.

[CLEI91] CLEIMAN,F.D., "Aspectos da evolução e difusão da IA e da Engenharia de Conhecimento no Brasil", *Relatório Técnico-Científico, COPPE-PADCT-CNPq*, 1991,8-38.

[COHE83] COHEN, M. e GROSSBERG, S., "Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks ", *IEEE Transactions on Systems, Man, and Cybernetics*, 1983, 815-825.

[COST86] COSTA,A.C.R., *Sobre os Fundamentos da Inteligência Artificial*, Relatório de Pesquisa n. 61, Universidade Federal do Rio Grande do Sul,1986.

[CROSS91] CROSS,M., "Japan's Quest for the Brainy Computer", *New Scientist*, January, 1991, 51-54.

[CARD90] CARDADOR,D.M.,*Representação de Conhecimento: Modelos Clássicos e Conexionistas*, Tese de Mestrado, Instituto Militar de Engenharia, 1990.

[DENI91] DENIS, F.A.R.M, *O Modelo Conexionista Evolutivo* , Tese de Mestrado, Programa de Engenharia de Sistemas e Computação, COPPE, UFRJ, 1991.

[DOLAN87] DOLAN, C. P. e DYER,M.G.,"Symbolic Schemata, Role Binding, and the Evolution of Structure in Connectionist Memories",*Proceeding of the First International Conference on Neural Networks*,San Diego, CA, 1987, V.II, 287-298.

[DOLAN88] DOLAN, C. P. e DYER,M.G.,"Parallel Retrieval of Concecptual Knowledge",*Proceeding of the 1988 Connectionist Summer School*,1988,273-280.

[DYER90] DYER,M.G., "Distribued Symbol Formation and Processing in Connectionist Networks", *J. Expt. Theor. Artif. Intell.*, vol.2,1990, 215-239.

[EBER90] EBERHART, R.C. e DOBBINS, R.W.,"Early Neural Network Development History: The Age of Camelot",*IEEE Engineering in Medicine and Biology*,Vol. 9 ,n.3., 1990,15-18.

[EFIM75] EFIMOV, N.V. e ROZENDORN, E.R., *Linear Algebra and Multidimensional Geometry*,Mir Publishers, Moscow,1975.

[FELD85] FELDMAN,J.A., "Four frames suffice: A provisional model of vision and space", *Behav. Brain. Sci.*, 8, 1985, 256-289.

[FERR89] FERREIRA,J.C., *Redes Neurais artificiais*, Texto do 5o. Simpósio de informática de Campina Grande, setembro, 1989.

[FODO88] FODOR,J.A., e PYLYSHYN,Z.W., "Connectionism and Cognitive architecture: A Critical Analysis", *Cognition*, 28, 3-71.

[GELD90] GELDER,T.V, "Compositionality: A Connectionist Variation on a Classical Theme" ,*Cognitive Science*, 14,1990, 355-384.

[GILM82] GILMORE,R.,*Lie Group, Lie Algebra and Some of their Application*, Wiley Interscience,1982.

[GREU78] GREUB,W.,*Multilinear Algebra*,Spring-Verlag, 2nd Edition,1978.

[GREU81] GREUB,W.,*Linear Algebra*,Spring-Verlag, 4th Edition,1981.

[HEBB61] HEBB,D.O., *Organization of Behavior*, Science Editions, Inc. New York, 1961.

[HECH88] HECHT-NIELSEN, R. "Applications of Counterpropagation Networks", *Neural Networks*, vol.1,1988,131-140.

[HINT80] HINTON,G.E.,*Draft of Technical Report*. University of California at San Diego,1980.

[HINT81] HINTON, G.E.,"Implementing Semantic Network in Parallel Hardware" , em G. E. HINTON e J. A. ANDERSON (Eds.), *Parallel Models of Associative Memory* , Hillsdate,N.J.,Lawrence Erlbaum, 1981.

[HINT86] HINTON, G,E., MCCLELLAND, J.L. e RUMELHART,D.E."Distribued Representations" , em D. E. RUMELHART e J. L. MCCLELLAND (Eds.), *Parallel Distributed Processing* (Vol.II), Cambridge, MA, The MIT Press, 1986.

[HINT88] HINTON, G.E. e MCCLELLAND, J.L. "Learning Representations by Recirculation", in Anderson (eds),*Neural Information Processing Systems*, (American Institute of Physics), New York,1988,358-367.

[HINT90] HINTON,G.E., "Mapping Part-Whole Hierarchies into Connectionist Networks", *Artificial Intelligence* ,46,1990,47-75.

[HOPF82] HOPFIELD,J.J.,"Neural Networks and Physical Systems with Emergent Collective Computational Abilities",*Proc. National Academy of Science,USA*, Vol.79, April 1982, 2554-2558.

[HOPF88] HOPFIElD, J.J., Allen, J. e Leighton,F.T., "Artificial Neural Networks", *Proc. of the Fifth MIT Conference - Advanced Research in VLSI* , Cambridge.

[WILB82] WILBER,K., *The Holographic Paradigm and Other Paradoxes*, Boulder, Colorado, Shambala, 1982.

[KIM90] KIM,S.,LEE,D. & Park,H.,"Implementation of Optical Holographic Heteroassociative Loop", *Conference Record of 1990 International Topical Meeting on Optical Computing*, Volume 1359, 270-272, Japan.

[KOHO87] KOHONE,T., "Adaptive , Associative , and Self-Organizing Function in Neural Computing", *Applied Optics*, 26, 1987, 4910-4918.

[KOSK85] KOSKO,B.,*Adaptative inference*, Verac, Inc. Technical Report.

[KOSK91] KOSKO,B., *Neural Networks and Fuzzy Systems - A Dynamical Systems approch to Machine Intelligence*, Prentice Hall, 1991.

[LANG71] LANG,S.,*Álgebra Linear*,Editora Universidade de Brasilia,1971.

[LASH50] LASHLEY, K.S. "In The Search of The Engram", *Society of Experimental Biology Symposium*, Num. 4, 1950, 478-505.

[LEGE90] LEGENDRE,L., MIYATA,Y. & SMOLENSKY, P., *Harmonic Grammar - A Formal Multi-Level Connectionist Theory of Linguistic Well-Formedness: Aplication*, American Institute of Physics, Technical Report CU-CS-464-90,Department of Computer Science, University of Colorado at Boulder,1990.

[LEGE90] LEGENDRE,L., MIYATA,Y. & SMOLENSKY, P., *Harmonic Grammar - A Formal Multi-Level Connectionist Theory of Linguistic Well-Formedness:*

Theoretical Foundations, American Institute of Physics, Technical Report CU-CS-465-90,Department of Computer Science, University of Colorado at Boulder,1990.

[LEGE90] LEGENDRE,L., MIYATA,Y. & SMOLENSKY, P., *Can Connectionism Contribute To Syntax ? Harmonic Grammar, with an Application*, American Institute of Physics, Technical Report CU-CS-485-90,Department of Computer Science, University of Colorado at Boulder,1990.

[LEGE91] LEGENDRE,L., MIYATA,Y. & SMOLENSKY, P., *Distributed Recursive Structure Processing*, American Institute of Physics, Technical Report CU-CS-514-91,Department of Computer Science, University of Colorado at Boulder,1991.

[LIPP87] LIPPMANN,R.P., "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, 1987, 4-22.

[LOOM68] LOOMIS,L.H. and STERNBERG,S., *Advanced Calculus*, Addison-Wesley, 1968, 305-320.

[MCCL86] MCCLELLAND, J. L.,KAWAMOTO, A.H."Mechanisms of Sentence Processing: Assigning Roles to Constituents", em D. E. RUMELHART e J. L. MCCLELLAND (Eds.), *Parallel Distributed Processing* (Vol.II), Cambridge, MA, The MIT Press, 1986.

[MCCL86] CCLELLAND, J. L.,ELMAN, J.L."Interative processes in speech perception: The TRACE model", em D. E. RUMELHART e J. L. MCCLELLAND (Eds.), *Parallel Distributed Processing* (Vol.II), Cambridge, MA, The MIT Press, 1986.

[MIIK91] MIIKKULAINEN,R.,DYER,M.G. "Natural Language Processing With Modular PDP Networks and Distributed Lexicon", *Cognitive Science*, Vol. 15, 3, 1991,343-400.

[MINS69] MINSKY, M.L. and PAPERT,S., *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge, Massachusetts,1969.

[MINS75] MINSKY,M.,"A Framework for Representing Knowlegde ", In P. Winston *The Psychology of Computer Vision*,NY: McGraw-Hill, 1975, 211-277.

[MINS91] MINSKY,M.,"Logical Analogical or Symbolic Versus Connectionist or Neat Versus Scruffy",*AI Magazine*, summer 1991,34-51.

[MURP91] MURPHY,J. "Tutorial: Neural Network Learning Algorithms " , *NASA/CCDS AV SPI - WNN-AIND 91*, 135-142.

[NEWE76] NEWELL,A. e SIMON,H.A., "Computer Science as Empirical Enquiry: Symbols and Search", *Communications of ACM*, Vol.19,3, Mar. 1976.

[NILS80] NILSSON,N.J., *Principles of Artificial Intelligence*, California,1980.

[PAO89] PAO.Y.-H., *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley Publishing Company,1989.

[PESS89a] PESSOA,L.A.,*Modelos Conexionistas e Representação do Conhecimento*, Dissertação de Graduação, Instituto de Matemática, UFRJ,1989.

[PESS89b] PESSOA, L. A. & REIS,A.,*Representação de Estruturas do Tipo Schemata em Modelos Conexionistas*, Relatório Técnico ES-207/89, Programa de Engenharia de Sistemas e Computação, COPPE, UFRJ, 1989.

[PESS90a] PESSOA, L. A.,*Aprendizado Não-Supervisionado em Redes Neuronais*, Tese de Mestrado, Programa de Engenharia de Sistemas e Computação, COPPE, UFRJ, 1990.

[PESS90b] PESSOA, L. A. & REIS,A., "Avaliação da Representação de Schemata em Modelos Conexionistas",*Revista Brasileira de Computação*, V.5,n.4,abr/jun,1990.

[POLL88] POLLACK, J.,"High-Level Connectionist Models", *AI Magazine*, winter 1988, 65-68.

[POLL90] POLLACK, J.B., "Recursive Distribued Representations", *Artificial Intelligence* , 46,1990,77-105.

[PRIB71] PRIBRAM,K.,*Languages of the Brain: Experimental Paradoxes and Principles in Neuropsychology*, Englewood Cliffs,N.J. Prentice Hall,1971.

[RICH88] RICH,E., *Inteligência Artificial*, McGraw-Hill,1988.

[ROSE85] ROSENFELD, R. e TOURETZKY, D., "A Survey of Coarse-Coded Symbol Memories", *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*: Vol I, 1985.

[ROSE88] ROSENFELD, R, e TOURETZKY, D., *Scaling Properties of Coarse-Coded Symbol Memories*, American Institute of Physics , 1988,652-661.

[RUME82] RUMELHART, D.E. e MCCLELLAND,J.L., "An interactive model of context context effects in letter perception, Part2: The contextual enhancement effect and some tests and extensions of the model",*Psychol. Rev.*,89, 1982, 60-94.

[RUME86] RUMELHART, D. E., HINTON, G. E., e MCCLELLAND, J. L., "A general framework for Parallel Distributed Processing", em D. E. RUMELHART e J. L. MCCLELLAND (Eds.),*Parallel Distributed Processing (Vol. I)*, Cambridge, MA, The MIT Press, 1986.

[SATA75] SATAKE,I.,*Linear Algebra*,Marcel Dekker, 1975.

[SCHA77] SCHANK, R.C. e Abelson, R.P.,*Scripts, Plans, Gols, and Understanding*, NJ: Erlbaum, 1977.

[SEJN87] SEJNOWSKI, T.J. e ROSENBERG, C.R., "Parallel networks that learn to pronounce English text",*Complex Systems*, 1,1987,145-168.

[SHAR91] SHARKEY,N.E., *Connectionist Representation Techniques*, Technical Report R 217, Department of Computer Science, University of Exeter,1991.

[SHAW82] SHAW, R., *Linear Algebra and Group Representations*, Academic Press Inc., vol.2, 1982.

[SIMP90] SIMPSON, P. K., *Artificial neural systems: Foundations, paradigms, applications, and implementations*, Elmsford, NY, Pergamon Press, 1990.

[SMIE92] ŚMIEJA, F.J.,MUHLENBEIN,H. *Reflective Modular Neural Network Systems*, Technical Report, German National Research Center for Computer Science (GMD),Germany,March 1992.

[SMOL87] SMOLENSKY, P., *On Variable Binding and the Representation of Symbolic Structures in Connectionist Systems*, Technical Report CU-CS-355-87, Department of Computer Science, University of Colorado at Boulder,1987.

[SMOL87b] SMOLENSKY, P., *A Method for Connectionist Variable Binding*, Technical Report CU-CS-356-87, Department of Computer Science, University of Colorado at Boulder,1987.

[SMOL88] SMOLENSKY, P., *Analysis of Distribued Representaaion of Constituent Structure in Connectionist Systems*, American Institute of Physics, 1988, 729-739.

[SMOL89] SMOLENSKY, P, "Tensor Product Production System: a Modular Architecture and Representation", *Connection Science*, Vol. 1, 1, 1989,53-68.

[SMOL90] SMOLENSKY, P., "Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems", *Artificial Intelligence* , 46,1990,159-216

[SMOL92] SMOLENSKY, P., *Principles for an Integrated Connectionist/Symbolic Theory of Higher Cognition*, Technical Report CU-CS-600-92, Department of Computer Science, University of Colorado at Boulder,1992.

[SOUC88] SOUCEK,B. e SOUCEK,M., *Neural and Massively Parallel Computers: The Sixth Generation*, Wiley-Interscience Publication, 1988.

[SOUC89] SOUCEK,B. *Neural and Concorrent Real-Time Systems: The Sixth Generation*, Wiley-Interscience Publication,1989.

[STYB88] STYBLINSKI,M.A. e MEYER,B.D.,"Fuzzy Cognitive Maps, Signal Flow Graphs, and Qualitative Circuit Analysis", *Proceedings of the 2nd IEEE International Conference on Neural Networks (ICNN-88)*, vol.II, 549-556, July 1988.

[TABE87] TABER,W.R. e SIEGEL,M., "Estimation of Expert Weight with Fuzzy Cognitive Map",*Proceeding of the 1st IEEE International Conference on Neural Networks (ICNN-87)*, vol.II, 319-325, June,1987.

[TOUR88] TOURETZKY,D. e HINTON, G., "A Distribued Connectionist Production System", *Cognitive Science*, 12, 1988,423-466.

[TOUR90] TOURETZKY,D., "BoltzCONS: Dynamic Symbol Structures in a Connectionist Network" ,*Artificial Intelligence*, 46,1990, 5-46.

[VALD91] VALDES, R.,"What is BioComputing",*Dr. Dobb's Journal*, V.175,n.4,April,1991.

[WASS89] WASSERMAN, P., *Neural Computing : Theory and Practice*, Van Nostrand Reinhold, 1989.

[WEST91] WEST,D.M. e TRAVIS,L.E., "From Society to Landscape: Alternative Metaphors for Artificial Intelligence", *AI Magazine*, summer 1991,69-83.

[WILL81] WILLSHAW,D.,"Holography, Associative Memory. and Inductive Generalization" , em HINTON, G.E. e ANDERSON, J.A. (Eds.), *Parallel Model of Associative Memory*, New Jersey, Lawrence Erlbaum Assoc3ates, 1981.

[ZHAN88] ZHANG,W., e CHEN,S.,"A Logical Architecture for Cognitive Maps",*Proceeding of the 2nd IEEE International Conference on Neural Networks (ICNN-88)*, vol. I,1988, 231-238.

Apêndice A

Implementações

A.1 Implementação da Representação Puramente Local

```
{#########################################################################
Programa para simular o Fuzzy Cognitive Map dado como exemplo em
KOSKO, Neural Networks and Fuzzy Systems - A Dynamical Systems
Approach to Machine Intelligence, Prentice Hall, 1991, 152-158.
UFRJ/COPPE/SISTEMAS - Pedro Paulo Ayrosa
#########################################################################}


program FCM;
type
matriz    = array [1..9] of
            array [1..9] of integer;
vetor     = array [1..9] of integer;
vetor_int = array [1..9] of integer;
var
  matrizs             : matriz;
  mapa                : matriz;
  estimulo            : vetor_int;
```

```
  atividade,saida     : vetor;
   n,i,j,k,lixo       : integer;

{#####################################################################
          O procedimento atribua_pesos monta a
            matriz W 9 x 9 de conexao causal.
#####################################################################}

procedure atribua_pesos;
begin
mapa[1,1] := 0;
mapa[1,2] := 1;
mapa[1,3] := 1;
mapa[1,4] := 0;
mapa[1,5] := 0;
mapa[1,6] := 0;
mapa[1,7] := 0;
mapa[1,8] := 1;
mapa[1,9] := 1;

mapa[2,1] := 1;
mapa[2,2] := 0;
mapa[2,3] := 1;
mapa[2,4] := 0;
mapa[2,5] := 0;
mapa[2,6] := 0;
mapa[2,7] := 0;
mapa[2,8] := 1;
mapa[2,9] := -1;

mapa[3,1] := 1;
mapa[3,2] := 0;
```

```
mapa[3,3] := 0;
mapa[3,4] := 1;
mapa[3,5] := 0;
mapa[3,6] :=-1;
mapa[3,7] := 0;
mapa[3,8] := 1;
mapa[3,9] := 1;

mapa[4,1] := 0;
mapa[4,2] := 0;
mapa[4,3] := 0;
mapa[4,4] := 0;
mapa[4,5] := 0;
mapa[4,6] := 1;
mapa[4,7] := 1;
mapa[4,8] := 0;
mapa[4,9] :=0 ;

mapa[5,1] := 0;
mapa[5,2] :=-1;
mapa[5,3] :=-1;
mapa[5,4] := 0;
mapa[5,5] := 0;
mapa[5,6] := 1;
mapa[5,7] := 1;
mapa[5,8] := 0;
mapa[5,9] := 0;

mapa[6,1] := 0;
mapa[6,2] := 0;
mapa[6,3] := 0;
mapa[6,4] := 1;
```

```
mapa[6,5] := 0;
mapa[6,6] := 0;
mapa[6,7] :=-1;
mapa[6,8] := -1;
mapa[6,9] := 0;

mapa[7,1] := 0;
mapa[7,2] := 0;
mapa[7,3] := 0;
mapa[7,4] := 0;
mapa[7,5] := 1;
mapa[7,6] := 0;
mapa[7,7] := 0;
mapa[7,8] :=-1;
mapa[7,9] := 0;

mapa[8,1] := 1;
mapa[8,2] := 0;
mapa[8,3] := 0;
mapa[8,4] := 0;
mapa[8,5] := 0;
mapa[8,6] := 0;
mapa[8,7] :=-1;
mapa[8,8] := 0;
mapa[8,9] := 0;

mapa[9,1] := 1;
mapa[9,2] := 0;
mapa[9,3] := 0;
mapa[9,4] := 0;
mapa[9,5] :=-1;
mapa[9,6] := 0;
```

```
mapa[9,7] := 0;
mapa[9,8] := 1;
mapa[9,9] := 0;
end;

{#####################################################################
                    Procedimento ver_saida
#####################################################################}

procedure ver_saida;
 begin
    for n:=1 to 9 do
      write(saida[n]:4);
    writeln;
    writeln;
 end;

{#####################################################################
                O procedimento limita_saida
              mantem o vetor dentro do limite (0,1)
#####################################################################}

procedure limita_saida;
  begin
     for n:=1 to 9 do
       begin
         saida[n] := saida[n] + estimulo[n];
         if saida[n] > 1 then saida[n]:= 1;
         if saida[n] < 1 then saida[n] := 0;
       end;
   end;
```

```
{#####################################################################
                    O procedimento zera_vetor
                    inicializa o vetor saida
#####################################################################}

procedure zera_vetor;
  begin
    for n:= 1 to 9 do
      saida[n]:= 0;
    end;

{#####################################################################
               O procedimento calcula_vetor
                   calcula o vetor saida
#####################################################################}

procedure calcula_vetor;
    begin
 for n:=1 to 9 do
   for i:=1 to 9 do
     saida[n] := saida[n]+estimulo[i] * mapa[i,n];
end;

{#####################################################################
        O procedimento inicio prepara realimentacao
#####################################################################}

procedure inicio;
begin
  for n:=1 to 9 do
     estimulo[n] := saida[n];
  zera_v'tor;
```

```
end;

{#####################################################################
                    O Programa Principal
#####################################################################}

begin

  atribua_pesos;
  zera_vetor;
  estimulo[1] := 1;
  estimulo[2] := 0;
  estimulo[3] := 0;
  estimulo[4] := 0;
  estimulo[5] := 0;
  estimulo[6] := 0;
  estimulo[7] := 0;
  estimulo[8] := 0;
  estimulo[9] := 0;
  calcula_vetor;
  ver_saida;
  limita_saida;
  ver_saida;
  for k:=1 to 4 do
     begin
       inicio;
       calcula_vetor;
       ver_saida;
       limita_saida;
       ver_saida ;
     end;
end.
```