

UNIVERSIDADE ESTADUAL PAULISTA – UNESP
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Reconhecimento de Padrões Lexicais por meio de Redes Neurais

Maurizio Babini

Orientador: Prof. Dr. Norian Marranghello

Dissertação apresentada à Faculdade de Engenharia de Ilha Solteira, UNESP – Câmpus de Ilha Solteira, para obtenção do título de Mestre em Engenharia Elétrica (Área de Concentração: Automação)

Ilha Solteira
Dezembro/2006

FICHA CATALOGRÁFICA

Elaborada pela Seção Técnica de Aquisição e Tratamento da Informação – Serviço Técnico de Biblioteca e Documentação da UNESP - Ilha Solteira.

B114r	<p>Babini, Maurizio Reconhecimento de padrões lexicais por meio de redes neurais / Maurizio Babini. -- Ilha Solteira : [s.n.], 2006 110 p. : il.</p> <p>Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira, 2006</p> <p>Orientador: Norian Marranghello Bibliografia: p. 108-110</p> <p>1. Redes neurais (Computação). 2. Reconhecimento de padrões. 3. Reconhecimento de padrões lexicais. 4. Rede di Kohonen. 5. Processamento da linguagem natural (Computação).</p>
-------	--

À minha filha Gabriella
À minha esposa Lidia
Aos meus Pais

Agradecimentos

À minha esposa Lídia e à minha filha Gabriella, pelo apoio que nos deram e nos dão para continuar a estudar e alcançar novas metas.

A meu orientador, o Prof. Dr. Norian Marranghello, pelos conselhos e orientação.

Aos Professores do Curso de Pós-Graduação em Engenharia Elétrica da UNESP de Ilha Solteira, pelas aulas e convivência no curso, e, em particular, ao Professor Aparecido Augusto de Carvalho.

Aos amigos e colegas do curso de Pós-Graduação em Engenharia Elétrica, pelas horas de estudo e pelos bons momentos passados dentro e fora das aulas.

Nil posse creari de nihilo
Lucrécio, *De rerum natura*

Sumário

Introdução	14
1 As Redes Neurais Artificiais.....	18
1.1 <i>O que são as Redes Neurais Artificiais?.....</i>	18
1.2 <i>Histórico das Redes Neurais Artificiais.....</i>	19
1.3 <i>Cérebro humano e neurônios biológicos.....</i>	21
1.4 <i>Neurônios artificiais.....</i>	25
1.5 <i>Funções de ativação.....</i>	30
1.6 <i>Arquiteturas de RNAs.....</i>	35
2 Mapas auto-organizáveis: o modelo de Kohonen.....	39
3 Reconhecimento de padrões e aprendizado.....	46
3.1 <i>Conceitos-chave.....</i>	46
3.2 <i>Inteligência Artificial “Simbólica” e “Subsimbólica” e aprendizado: definições preliminares.....</i>	49
3.3 <i>Métodos de aprendizado e classificação de padrões.....</i>	51
4 A transformação de textos em vetores.....	56
4.1 <i>Modelo Booleano.....</i>	57
4.2 <i>Modelo de espaço vetorial (Vector-space model – VSM).....</i>	61
4.2.1 <i>Similaridade entre vetores.....</i>	63
4.2.2 <i>Importância das palavras e cálculo dos pesos: considerações gerais.....</i>	66
4.2.3 <i>Frequência normalizada de um termo (term frequency – tf) k_i em d_j.....</i>	67
4.2.4 <i>Frequência inversa de uma palavra em um documento (Inverse document frequency).....</i>	69
4.2.5 <i>O cálculo dos pesos.....</i>	70
5 O modelo semântico de Pottier: os conceitos de sema, conceito, e noema.....	72
5.1 <i>Os percursos onomasiológico e semasiológico.....</i>	73
5.2 <i>Sema, conceito e noema no modelo de signo de Pottier.....</i>	76

6	A implementação da rede.....	84
6.1	<i>O programa Nenet.....</i>	84
6.2	<i>Da análise semântica aos padrões.....</i>	85
6.3	<i>Testando a rede.....</i>	97
	Conclusões	105
	Referências	108

Lista de figuras

Figura 1.1. – Componentes principais de um neurônio biológico. (LEVINE; SHEFNER, 1991).....	23
Figura 1.2. – Esquema de uma sinapse. A - Neurônio transmissor B - Neurônio receptor 1. Mitocôndria 2. Junção comunicante (feita de conexinas). 3. Sinal elétrico. (WIKIPEDIA, 2006)	23
Figura 1.3. – Potencial de ação em um neurônio. (BRAGA et al., 1998, p.7).....	24
Figura 1.4. – Neurônio de McCulloch e Pitts. (BRAGA et al., 1998, p.9).	26
Figura 1.5. – (a) Neurônio biológico e (b) neurônio artificial.....	27
Figura 1.6. – Modelo não-linear de um neurônio (HAYKIN, 2001, p.36).	27
Figura 1.7. – Efeito do <i>bias</i> sobre o campo local induzido (HAYKIN, 2001, p.37).....	29
Figura 1.8. – Modelo de neurônio com <i>bias</i> como entrada fixa (HAYKIN, 2001, p.37).....	29
Figura 1.9. – Função Linear (BRAGA et al., 1998, p.10).....	31
Figura 1.10. – Função Rampa.....	31
Figura 1.11. – Função de Limiar (a) (HAYKIN, 2001, p.41)	32
Figura 1.12 – Função de Limiar (b).....	33
Figura 1.13 – Função sigmóide (a). Gráfico correspondente à Eq. (1.11). (HAYKIN, 2001, p.40), (BRAGA et al., 1998, p.10).....	34
Figura 1.14. – Função sigmóide (b). Gráfico correspondente à Eq. (1.12).....	34
Figura 1.15. – Rede de camada única, <i>feedforward</i> (alimentada adiante), completamente conectada. (HAYKIN, 2001, p.47).....	35
Figura 1.16. – Rede multicamadas, <i>feedforward</i> , completamente conectada (HAYKIN, 2001, p.48)	36
Figura 1.17. – Rede de múltiplas camadas, <i>feedback</i> e parcialmente conectada (BRAGA et al., 1998, p.12).....	37

Figura 1.18 – Rede de múltiplas camadas, <i>feedforward</i> e parcialmente conectada (BRAGA et al., 1998, p.12)	38
Figura 2.1. – Estrutura da rede neural de Kohonen. (MINUSSI, 2005)	40
Figura 2.2. Conexão do vetor padrão de entrada X com o j-ésimo neurônio da rede de Kohonen. (MINUSSI, 2005)	41
Figura 2.3.– Mapeamento de vetores em uma rede de Kohonen. (WANGENHEIM, 2006)	42
Figura 2.4. – Forma da função de vizinhança. (WANGENHEIM, 2006)	44
Figura 3.1. – Etapas de um projeto sobre reconhecimento de padrões (MARQUES de SÁ, 2006)	47
Figura 3.2.– Aprendizado de máquina (WANGENHEIM, 2006)	52
Figura 3.3.– Forma de apresentação dos padrões a serem aprendidos: aprendizado supervisionado e aprendizado não-supervisionado (WANGENHEIM, 2006)	53
Figura 3.4. – Modelos de inferência utilizados no aprendizado (WANGENHEIM, 2006).....	54
Figura 3.5. – Objetivo e critérios de aprendizado (WANGENHEIM, 2006).....	55
Figura 4.1. – Representação gráfica dos três componentes conjuntivos da consulta	58
Figura 5.1.– Percurso onomasiológico. (POTTIER, 1992, p. 16)	73
Figura 5.2. – Percurso semasiológico. (POTTIER, 1992, p.17).....	74
Figura 5.3. – Percurso onomasiológico. (POTTIER, 1992, p.18)	75
Figura 5.4. – Representações gráficas de alguns noemas. (POTTIER, 1992, p.78).....	77
Figura 5.5. – Representação gráfica de alguns noemas. (POTTIER, 1992, p.17).....	78
Figura 5.6. – O modelo de signo lingüístico de Pottier. (POTTIER, 1985, p.26).....	79
Figura 5.7. – Plano do significado e plano do significante. (POTTIER, 1985, p.77).....	80
Figura 5.8. – O semema: semantema, classema e virtuemema. (POTTIER, 1985, p.30)	81
Figura 6.1 – Escolha dos parâmetros de inicialização da rede	87
Figura 6.2 – Representação gráfica da rede inicializada.....	88
Figura 6.3. – Valores dos pesos sinápticos do vetor do primeiro neurônio da rede.	89
Figura 6.4. – Rede de 6 x 6 neurônios criada com o programa SNNS.	90
Figura 6.5. – Escolha dos parâmetros para treinar a rede	91
Figura 6.6. – Representação gráfica da rede depois do treinamento	92
Figura 6.7. – Escolha dos parâmetros para testar a rede	93
Figura 6.8. – Representação gráfica da rede depois da fase de teste	94
Figura 6.9. – Representação gráfica do primeiro parâmetro dos vetores de teste	95

Figura 6.10. – Representação gráfica do sexto parâmetro dos vetores de teste	96
Figura 6.11. – Representação gráfica do segundo e do quarto parâmetro dos vetores de teste.....	97
Figura 6.12. – Representação gráfica da rede depois da apresentação do padrão desconhecido.....	98
Figura 6.13. – Representação gráfica da rede depois da apresentação de padrões.....	100
Figura 6.14 – Representação gráfica da rede depois da apresentação de padrões corrompidos em todos os parâmetros.	102
Figura 6.15 – Representação gráfica da rede depois da apresentação de padrões corrompidos no parâmetro 1	104

Lista de tabelas

Tabela 4.1 Conceitos básicos e símbolos utilizados para descrever o modelo booleano. (KAESTNER, 2006), (BAEZA-YATES; RIBEIRO-NETO, 1999, p.25)	57
Tabela 4.2 Conceitos básicos e símbolos utilizados para descrever o modelo de espaço vetorial. (ACKERMAN, 2003)	62
Tabela 4.3 Critérios para atribuição de pesos. (ACKERMAN, 2003)	67
Tabela 4.4. Variação da frequência normalizada. (ACKERMAN, 2003)	68
Tabela 4.5 Variação da frequência inversa. (ACKERMAN, 2003)	69
Tabela 4.6 Variação do peso em função da frequência normalizada e da frequência inversa. (ACKERMAN, 2003).....	70
Tabela 5.1. – Categorias de conceitos gerais. (POTTIER, 1992, p.72).....	76
Tabela 5.2. Análise semântica de um conjunto de palavras. (POTTIER, 1964, p.125)	82
Tabela 6.1 Transformação em vetores de um conjunto de palavras.....	86
Tabela 6.2. Vetores de inicialização, treinamento e de teste.....	86
Tabela 6.3. – Vetor de teste de padrão desconhecido	98
Tabela 6.4. – Vetores de teste de padrões corrompidos no parâmetro S_6	99
Tabela 6.5 – Vetor de teste com padrões corrompidos em todos os parâmetros.....	101
Tabela 6.6 – Vetor de teste com padrões muito corrompidos no parâmetro S_6	103

Resumo: A compreensão da linguagem humana é uma das tarefas mais difíceis do Processamento da Linguagem Natural (PLN) e de modo mais geral da Automação e da Inteligência Artificial (IA). O objetivo desta pesquisa é estudar os mecanismos que permitem utilizar uma rede neural artificial para poder interpretar textos. Este trabalho deveria ser utilizado, futuramente, para criar uma interface em um ambiente de co-projeto, capaz de agrupar/classificar termos/conceitos, reconhecendo padrões textuais. Para alcançar nossos objetivos de pesquisa em nível de Mestrado, utilizamos o modelo semântico de Bernard Pottier, e uma Rede Neural Artificial de Kohonen. A escolha do modelo de Bernard Pottier deve-se ao fato de que este autor é um dos mais conceituados lingüistas da atualidade e que seu modelo é largamente utilizado por pesquisadores de vários países, tendo sido, assim, comprovada a sua validade. No que diz respeito à rede de Kohonen, acreditamos que seja a mais indicada para este tipo de aplicação, tendo em vista o fato de que essa rede tenta imitar o funcionamento do cérebro humano, em particular, reproduzindo o mapeamento de suas áreas especializadas, e tendo como hipótese de partida que, no córtex humano, conceitos similares ou de áreas afins distribuem-se em áreas limítrofes. A escolha desse tipo de rede para o nosso trabalho deve-se, outrossim, ao fato de que ela utiliza um tipo de treinamento competitivo e não-supervisionado que permite organizar os vetores (dados) de entrada em agrupamentos (*clusters*).

Palavras-chave: Redes Neurais Artificiais; Reconhecimento de Padrões; Reconhecimento de Padrões Lexicais; Reconhecimento de Padrões Textuais; Rede de Kohonen; Ambiente de Co-projeto.

Lexical Pattern Recognition using Artificial Neural Networks

Abstract: The understanding of human language is one of the most difficult tasks of Natural Language Processing (NLP), and, in general, of Automation and Artificial Intelligence (AI). The aim of our research is to study the mechanisms that allow using an artificial neural network for interpreting text. Later, our work should be used to create an interface, in a hardware/software co-design environment, capable of clustering/classifying terms/concepts, and recognizing text patterns. In order to achieve the objectives of our research, we used the semantic model of Bernard Pottier, and a Kohonen Artificial Neural Network. The choice of Bernard Pottier's model was motivated by the fact that the author is one of the most eminent linguists nowadays, and his model is largely used by researchers in many countries, thus proving the validity of his proposal. About the Kohonen net, we believe that it is the most appropriate net for this kind of application, due to the fact that this net tries to imitate the functioning of the human brain, particularly reproducing the map of its specialized areas, as well as due to the fact that this net has as initial hypothesis that, in the human cortex, similar concepts or concepts of similar areas are distributed in closed areas. Another reason for the choice of this kind of net in our study is that it uses a competitive and non-supervising training, that allows organizing entry vectors (data) in clusters.

Keywords: Artificial Neural Networks; Pattern Recognition; Text Pattern Recognition; Lexical Pattern Recognition; Kohonen net; Co-design Environment.

Introdução

Reconhecer padrões é uma das atividades mais importantes para o desenvolvimento do ser humano, e lhe permite de agrupar e classificar os objetos do mundo. Reconhecer padrões é identificar sons, imagens, cheiros, e, também reconhecer palavras, frases e interpretar textos. No entanto, reconhecer padrões de tipo textual, em outras palavras, compreender a linguagem natural, é uma das tarefas mais difíceis para o cérebro humano, tarefa que se torna ainda mais difícil quando deve ser realizada por uma máquina. O Processamento da Linguagem Natural (PLN) e de modo mais geral a Automação e a Inteligência Artificial (IA) tentam achar os meios para realizar essa tarefa. Dentre eles, o que nos pareceu mais indicado para a compreensão da linguagem natural foram as Redes Neurais Artificiais.

Objetivo dessa pesquisa é estudar os mecanismos que permitem utilizar uma rede neural artificial para poder reconhecer padrões de tipo textual. Nossos estudos serão utilizados, futuramente, para realizar uma interface, em um ambiente de co-projeto, capaz de agrupar termos/conceitos, e em seguida classifica-los, reconhecendo padrões de tipo textual.

Na Figura 1 são especificadas as diferentes etapas de um co-projeto, que tem como objetivos principais otimizar o desenvolvimento de sistemas e reduzir tempo e custo de projeto (Marranghello, 2005).

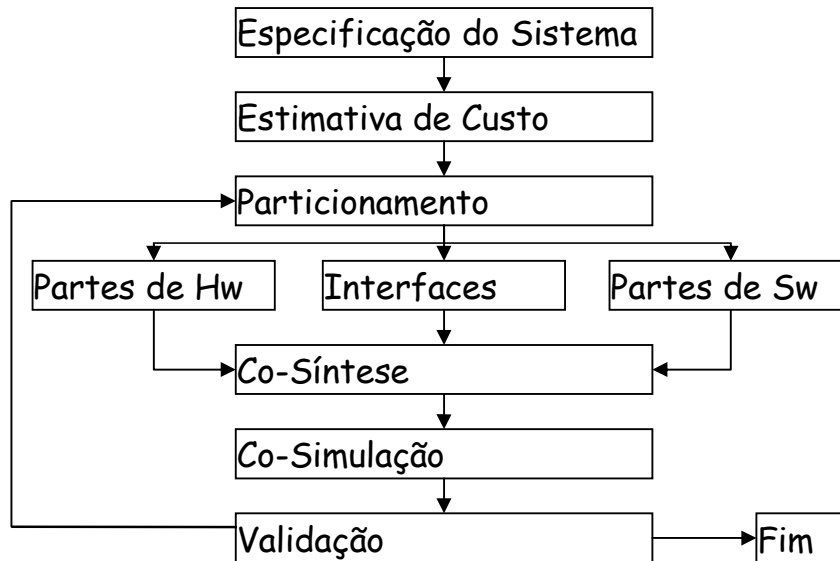


Figura 1 – Etapas para realização de um sistema em co-projeto

Na figura 2 se mostra o papel que a nossa interface deverá desenvolver:

Usuário:

Preciso de um sistema que agrupe/classifique termos a partir de sua descrição semântica.

Interface:

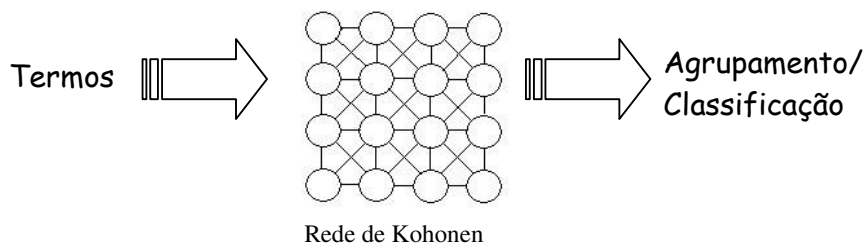


Figura 2 – Funcionamento esquemático da interface

Para alcançar nossos objetivos de pesquisa utilizamos um modelo proveniente da lingüística, o modelo semântico do pesquisador francês Bernard Pottier, e um tipo de Rede Neural Artificial, a do finlandês Teuvo Kohonen.

O modelo de Bernard Pottier foi escolhido em função do fato de que é amplamente utilizado nos estudos lingüísticos, tendo seu valor reconhecido por pesquisadores de muitos países, e por suas características intrínsecas. O modelo semântico de Pottier é um dos mais eficazes para a descrição e a compreensão da linguagem natural. Pottier utiliza rigor matemático em suas descrições lingüísticas, e seu modelo de análise semântica, nos pareceu o mais indicado para a transformação em vetores/padrões de termos/conceitos.

No que diz respeito à rede de Kohonen, muitas foram as razões que nos levaram a escolha desse tipo de rede. Desde os anos setenta, este pesquisador estuda os mecanismos do aprendizado humano tentando reproduzir um modelo de rede neural artificial que imite o cérebro humano. A rede de Kohonen baseia-se nas teorias sobre o funcionamento do cérebro humano, segundo as quais, conceitos similares são aprendidos e agrupados em áreas próximas, sendo assim, possível, um mapeamento do cérebro em áreas especializadas. Para realizar essa tarefa, a rede de Kohonen utiliza um tipo de treinamento competitivo e não supervisionado, que permite agrupar os vetores de entrada, e ser utilizada, em seguida, como classificador. Além disso, a Rede de Kohonen tem um bom desempenho em presença de ruído, reconhecendo padrões corrompidos.

No Capítulo 1, estudaremos os conceitos-chave das *Redes Neurais Artificiais* (RNAs) começando por um breve histórico das RNAs. Em seguida procederemos a uma comparação entre um modelo de neurônio biológico e de um neurônio artificial. Estudaremos as principais estruturas de RNAs, mostrando os elementos de processamento, as diferentes funções de ativação e a conectividade das redes, bem como alguns exemplos de arquiteturas de RNAs.

No Capítulo 2, estudaremos os *mapas auto-organizáveis* do finlandês Teuvo Kohonen. Analisaremos do ponto de vista matemático os três processos fundamentais que descrevem o funcionamento de uma Rede de Kohonen, isto é, o processo competitivo, o processo cooperativo e o processo adaptativo sináptico.

No Capítulo 3, estudaremos alguns conceitos-chave da *Inteligência artificial*, começando pelas definições de padrão e de reconhecimento de padrões, de *Inteligência Artificial Simbólica e Subsimbólica* e de aprendizado. Em seguida analisaremos os principais métodos de aprendizado e de classificação de padrões.

Tendo em vista o fato de que nosso objetivo é utilizar uma rede neural de Kohonen para reconhecimento de padrões lexicais, no Capítulo 4, estudaremos dois métodos clássicos de Recuperação de Informações: o *modelo booleano* e o *modelo de espaço vetorial*. Esses dois métodos nos ajudaram a compreender como transformar textos em vetores e como comparar entre si tais vetores. Analisaremos de maneira mais aprofundada o segundo, devido a suas semelhanças com a rede de Kohonen.

No Capítulo 5, analisaremos o modelo semântico de Bernard Pottier, partindo das definições de percurso semasiológico e de percurso onomasiológico, que nos permitirão melhor entender como o lingüista explica os processos de codificação e decodificação da mensagem. Em seguida, estudaremos os significados de sema, conceito, e noema, que são os elementos constitutivos do significado de uma unidade lexical. Por fim, veremos como é definido um signo lingüístico por Pottier e o modelo de análise semântica proposto por ele, analisando o exemplo clássico de *cadeira, poltrona, banquinho, sofá, pufe e assento*.

No Capítulo 6, implementaremos uma Rede de Kohonen composta de trinta e seis neurônios (6 x 6) por meio do *software* Nenet 1.1., do Nenet Team, da Universidade de Tecnologia de Helsinski. O modelo de análise semântica proposto por Pottier será utilizado como ponto de partida para constituir um pequeno conjunto de vetores a serem submetidos à Rede. Inicializaremos, treinaremos e testaremos a Rede com esse conjunto de vetores para analisar seu comportamento.

As conclusões terminarão nosso trabalho de pesquisa.

1 As Redes Neurais Artificiais

Este capítulo tem como objetivo estudar as Redes Neurais Artificiais (RNAs) e suas estruturas mais apropriadas para a utilização no reconhecimento de padrões. Nas páginas que se seguem, abordaremos aspectos históricos das Redes Neurais (suas origens e desenvolvimento), procederemos a comparação entre um modelo de neurônio biológico e de um neurônio artificial. Analisaremos ainda as principais estruturas de Redes Neurais Artificiais, mostrando os elementos de processamento, a conectividade e os tipos de funções de ativação. Concluiremos este capítulo com a análise de alguns exemplos de arquiteturas de RNAs.

1.1 O que são as Redes Neurais Artificiais?

Segundo Braga et al. (1998, p. 1), as Redes Neurais Artificiais, também conhecidas como redes conexionistas e sistemas de processamento paralelo e distribuído, “são sistemas paralelos distribuídos compostos por unidades de processamento simples (nodos) que computam determinadas funções matemáticas (normalmente não-lineares)”. As unidades que compõem as redes neurais podem ser “dispostas em uma ou mais camadas e interligadas por um grande número de conexões, geralmente unidirecionais” (BRAGA et al., 1998, p. 1). Na maioria das redes, às conexões são associados pesos, com diferentes funções, dentre elas a de controlar ou modificar a entrada de cada neurônio da rede e a de armazenar o conhecimento adquirido pela rede. E ainda, “esta forma de computação não algorítmica é caracterizada por

sistemas que, em algum nível, relembram a estrutura do cérebro humano” (BRAGA et al., 1998, p. 1).

Segundo Haykin (2001, p.28), uma rede neural “se assemelha ao cérebro humano em dois aspectos”:

O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem;

Forças de conexão entre os neurônios (pesos sinápticos) são utilizadas para armazenar o conhecimento adquirido.

Devido a essas semelhanças, as estruturas das redes neurais são criadas inspirando-se diretamente nas estruturas do cérebro humano, que, no entanto, continua sendo bem mais complexo do que a mais complexa das redes neurais existentes. O que torna as redes neurais artificiais extremamente interessantes aos pesquisadores de muitas áreas é justamente sua capacidade de aprender.

As redes neurais artificiais são estudadas e utilizadas em áreas bem diferentes e distantes uma das outras tais como: Ciências da Computação, Economia, Engenharia, Matemática, Medicina e Psicologia. Na área da Engenharia, encontram inúmeras aplicações na Inteligência Artificial, sendo utilizadas, sobretudo, no Reconhecimento de Padrões.

1.2 Histórico das Redes Neurais Artificiais

Os primeiros estudos sobre as RNAs começaram em 1943, com Warren McCulloch (psiquiatra e neuro-anatomista) e Walter Pitts (matemático), que realizaram o primeiro modelo de neurônio artificial, baseando-se na estrutura do cérebro humano e no funcionamento dos neurônios biológicos. Segundo Braga et al. (BRAGA et al., 1998, p.2), no artigo publicado em 1943 “*A Logical Calculus of the Ideas Immanent In Nervous Activity*” (McCULLOCH; PITTS, 1943) “é apresentada uma discussão sofisticada de redes lógicas de nodos e novas idéias sobre máquinas de estados finitos”.

O trabalho “*The organization of behavior*” de Donald Donald Hebb (HEBB, 1949) foi o ponto de partida para o desenvolvimento dos algoritmos de aprendizado das RNAs

(WASSERMAN, 1989, p.4). Segundo Braga et al. (BRAGA et al., 1998, p.2), Hebb “mostrou que a aprendizagem de redes neurais é conseguida através da variação dos pesos de entrada dos nodos”. A teoria proposta por Hebb, hoje conhecida como *Regra de Hebb*, “baseada no reforço das ligações sinápticas entre nodos excitados”, é ainda “hoje utilizada em vários algoritmos de aprendizado” (BRAGA et al., 1998, p.2).

Em 1958, Frank Rosenblatt criou um novo modelo de neurônio artificial, chamado *perceptron*, e propôs um algoritmo para treinar a rede (BRAGA et al., 1998, p. 2). O *perceptron* mais simples proposto por esse autor possui, segundo Braga et al. (BRAGA et al., 1998, p.2) três camadas: “a primeira recebe as entradas do meio e possui conexões fixas (retina), a segunda recebe impulsos da primeira através das conexões, cuja eficiência de transmissão (peso) é ajustável e, por sua vez, envia saídas para a terceira camada (resposta)”. O *perceptron* proposto por Rosenblatt atua como classificador, reconhecendo padrões e “dividindo o espaço de entrada em regiões distintas para cada uma das classes existentes” (BRAGA et al., 1998, p.2).

Em 1960, Widrow e Hoff criaram uma nova regra de aprendizado, conhecida como *regra de Widrow-Hoff* ou *regra delta*. Esta se baseia “no método do gradiente para minimização do erro na saída de um neurônio com resposta linear” (BRAGA et al., 1998, p.3). Ainda hoje a Regra Delta é bastante utilizada (BRAGA et al., 1998, p.3).

Em 1969, Minsky e Papert apontaram algumas falhas das RNAs e algumas tarefas que o *perceptron* não podia desempenhar, como “detectar paridade, conectividade e simetria, que são problemas não-linearmente separáveis” (BRAGA et al., 1998, p.2).

A década de 1970 foi um período de quase-estagnação dos estudos sobre as redes neurais, devido ao impacto dos trabalhos de Minsky e Papert (BRAGA et al., 1998, p.3) e à falta de verbas para pesquisas sobre redes neurais artificiais (HAYKIN, 2001, p.66). Destacou-se, no entanto, o trabalho de alguns pesquisadores, como o finlandês Teuvo Kohonen com seus estudos sobre as memórias associativas (Kohonen, 1974).

Graças ao trabalho de John Hopfield, que, em 1982, publicou o artigo “*Neural network and physical system with emergent collective properties*” (HOPFIELD, 1982), que teve grande impacto sobre a comunidade científica, os estudos sobre as RNAs tiveram uma nova retomada. Hopfield mostrou a “relação entre as redes recorrentes auto-associativas e sistemas físicos” (BRAGA et al., 1998, p.3). Outro marco na história das RNAs é a publicação do artigo de Kohonen sobre os mapas auto-organizáveis “*Self-organized formation of*

topologically correct feature maps” (KOHONEN, 1982), que propõe um novo tipo de rede baseado em uma rede unidimensional ou bidimensional (HAYKIN, 2001, p.67).

Em 1986, Rumelhart, Hinton e Williams divulgam o desenvolvimento do algoritmo de retropropagação (*back-propagation*), utilizado para a aprendizagem por retropropagação. Tal algoritmo tornou-se rapidamente o mais utilizado para o treinamento de perceptrons de múltiplas camadas (HAYKIN, 2001, p.68).

Depois da difusão do algoritmo de retropropagação, foi descoberto que “tal algoritmo havia sido descrito anteriormente por Werbos na sua tese de doutorado na Universidade de Harvard em agosto de 1974” (HAYKIN, 2001, p.68).

Em 1988, Broomhead e Lowe descreveram um novo tipo de redes *feedforward* utilizando *funções de base radial* (RBF – *radial basis function*), ainda hoje muito utilizadas para aplicações com aprendizado de tipo supervisionado.

No início dos anos 90 foi inventado por Vapnik e outros autores um novo tipo de redes neurais de aprendizado supervisionado, chamado de *máquina de vetor de suporte*, adequado para utilização em reconhecimento de padrões, regressão e problemas de estimação de densidade (HAYKIN, 2001, p.69).

As redes neurais artificiais continuaram na década de 90 e continuam, ainda hoje, a se desenvolver. Novos modelos de neurônios foram propostos e as redes neurais foram utilizadas com sucesso em inúmeras aplicações. Dentre as áreas de aplicação mais promissoras das redes neurais podem ser citadas a da Inteligência Artificial e a do Processamento de Linguagem Natural, áreas em que desenvolvemos nosso trabalho.

1.3 Cérebro humano e neurônios biológicos

A estrutura do sistema nervoso tornou-se mais conhecida graças aos esforços de dois médicos e histologistas, o espanhol Santiago Ramón y Cajal (1852-1934) e o italiano Camillo Golgi (1843-1926) que, em 1906 receberam, conjuntamente, o Prêmio Nobel de Medicina.

Graças a um método de coloração histológica à base de nitrato de prata, inventado por Golgi, Ramón y Cajal “postulou que o sistema nervoso é composto por bilhões de neurônios

distintos e que estas células se encontram polarizadas. Cajal sugeriu que os neurônios, em vez de formarem uma teia contínua, comunicam entre si através de ligações especializadas chamadas sinapses” (WIKIPEDIA, 2006).

Estima-se que o cérebro humano contenha cerca de “dez bilhões de neurônios no córtex e sessenta trilhões de sinapses ou conexões” (HAYKIN, 2001, p.33). O que impressiona no cérebro humano, além da quantidade de neurônios, é o número extremamente elevado de conexões.

Algo comparável, em complexidade, à estrutura do cérebro humano poderia ser o sistema da rede telefônica dos Estados Unidos de América, embora, com as novas tecnologias, os fios e as conexões tendam a diminuir, deixando espaço a transmissores de rádio e telefones sem fio. As RNAs também tentam reproduzir o funcionamento do cérebro humano, no entanto, nenhuma rede neural artificial conseguiu ainda alcançar, em complexidade e tamanho, o funcionamento do cérebro humano.

O neurônio artificial, elemento constitutivo de base das redes neurais artificiais, tem como modelo o neurônio biológico. No entanto, ainda hoje, o que se sabe do funcionamento do neurônio biológico e do cérebro humano é bem pouco. Uma boa descrição dos neurônios biológicos pode ser encontrada em um manual médico de Histologia, disciplina da Medicina que estuda os diferentes tipos de tecidos biológicos.

As partes principais do neurônio são o *corpo celular* ou *soma*, os *dendritos*, o *axônio* e o *telodendro*.

O *corpo celular* é a parte central do neurônio, contém o *núcleo (celular)* e é responsável pela geração dos impulsos nervosos.

Os *dendritos* são ramificações que partem do corpo celular do neurônio e que têm como função receber os sinais nervosos.

O *axônio* é um prolongamento do neurônio responsável pela propagação de impulsos nervosos que partem do corpo celular. É envolto pela *bainha de mielina*, que tem como função isolar eletricamente os axônios, permitindo uma propagação mais rápida dos impulsos nervosos. Perdas de mielina têm conseqüências catastróficas no funcionamento do cérebro, que se torna mais lento e incapaz de controlar várias funções do organismo humano.

O *telodendro* é constituído pelas ramificações terminais do axônio.

A Figura 1.1, mostra os componentes principais de um neurônio biológico:

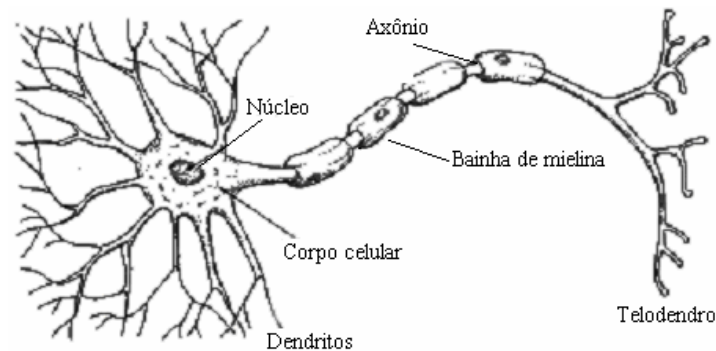


Figura 1.1. – Componentes principais de um neurônio biológico. (LEVINE; SHEFNER, 1991)

Os espaços entre os dendritos de um neurônio e o axônio de outro são chamados de *sinapses*. A transmissão de um impulso nervoso de um neurônio para outro passa de um axônio para os dendritos de outro neurônio por meio de neurotransmissores, substâncias químicas que são produzidas na célula emissora. Segundo Braga et al. (1998, p.6), “a comunicação no cérebro pode ser de dois tipos: sinais químicos através das sinapses e sinais elétricos dentro do neurônio. A ação da membrana dos neurônios é que cria a habilidade de produzir e transmitir esses sinais. A membrana que envolve o exterior do corpo do neurônio tem a capacidade de gerar impulsos nervosos (elétricos)”. Esses sinais são combinados no corpo celular “e, se o valor resultante for acima do limiar de excitação do neurônio, um impulso elétrico é produzido e propagado através do axônio para os neurônios seguintes” (BRAGA et al., 1998, p.6). Na Figura 1.2, é indicado o funcionamento de uma sinapse:

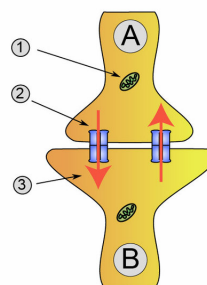


Figura 1.2. – Esquema de uma sinapse. A - Neurônio transmissor B - Neurônio receptor 1. Mitocôndria 2. Junção comunicante (feita de conexinas). 3. Sinal elétrico. (WIKIPEDIA, 2006)

A geração de impulsos elétricos se realiza porque “há uma diferença de potencial entre o interior e o exterior do neurônio, ocasionada pela diferença entre a concentração de potássio (interna à célula) e sódio (externa a célula)” (BRAGA et al., 1998, p.6). A concentração de íons de potássio contidos na célula gera “um potencial elétrico de -70mV (*potencial de repouso*) em relação ao exterior” (BRAGA et al., 1998, p.6).

Para que a célula dispare, “produzindo um potencial de ação (impulso nervoso), é preciso que os impulsos das sinapses reduzam este nível para cerca de -50mV ” (BRAGA et al., 1998, p.6). Quando o potencial é reduzido, o “fluxo de sódio e potássio é invertido, e o interior da célula torna-se subitamente positivo em relação ao exterior” (BRAGA et al., 1998, p.6). Essa inversão de polaridade permite a propagação do impulso nervoso no axônio até as ramificações do telodendro, onde são produzidos os neurotransmissores que, através das sinapses, permitem a propagação de um impulso nervoso até outro neurônio.

No gráfico da Figura 1.3 é representada a produção de um impulso nervoso em um neurônio.

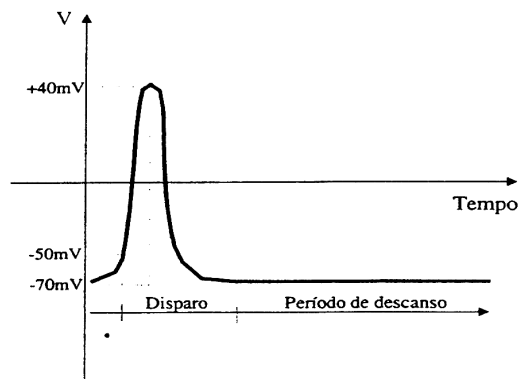


Figura 1.3. – Potencial de ação em um neurônio. (BRAGA et al., 1998, p.7).

Depois do disparo, “o neurônio entra em um período de refração (período em que o axônio não pode ser estimulado), durante o qual ele retorna ao seu potencial de repouso, preparando-se para a geração de um novo impulso” (BRAGA et al., 1998, p.7).

Comparando a velocidade de funcionamento de um neurônio biológico e de um neurônio artificial, constatamos que, no primeiro, uma operação acontece na ordem de

milissegundos (10^{-3} s), enquanto que a ordem de grandeza de uma operação em um circuito eletrônico é de nanossegundos (10^{-9} s) (HAYKIN, 2001, p.33). A relativa lentidão do cérebro humano em relação a uma rede neural artificial é compensada pela estrutura do cérebro e pelo número de sinapses e conexões, que permitem ao cérebro humano um altíssimo desempenho.

No que diz respeito à eficiência energética do cérebro humano e de um computador, cabe ressaltar que o primeiro gasta bem menos energia por operação, i.e., 10^{-16} J por operação por segundo, enquanto que um bom computador utiliza bem mais: cerca de 10^{-6} J por operação por segundo (HAYKIN, 2001, p.32).

É interessante ressaltar que há muitas semelhanças entre o funcionamento de um computador e de um cérebro humano no que diz respeito à lógica binária, e ao tipo de *hardware* utilizado para realizar um computador. Essas analogias de funcionamento são evidentes desde o começo da era da Informática, como no caso das memórias de núcleos magnéticos e mais recentemente dos *flip-flops*. No entanto, contrariamente aos estudos sobre as redes neurais artificiais, que tiveram como ponto de partida os estudos sobre o cérebro humano, os primeiros estudos sobre o computador foram desenvolvidos de maneira autônoma, embora o *flip-flop* tenha um funcionamento bem parecido com o neurônio biológico do cérebro humano.

1.4 Neurônios artificiais

O neurônio artificial é o elemento constitutivo de base de uma rede neural. O funcionamento do primeiro neurônio artificial, proposto por McCulloch e Pitts em 1943, era baseado nos estudos da época sobre os neurônios biológicos. Assim, o neurônio artificial tem terminais de entrada (x_1, x_2, \dots, x_n), que têm a função dos dendritos do neurônio biológico, um único terminal de saída (y), parecido ao axônio (do neurônio biológico) e um somador central mais uma função de ativação, que podem ser comparados ao corpo celular (do neurônio biológico), como podemos ver na Figura 1.4:

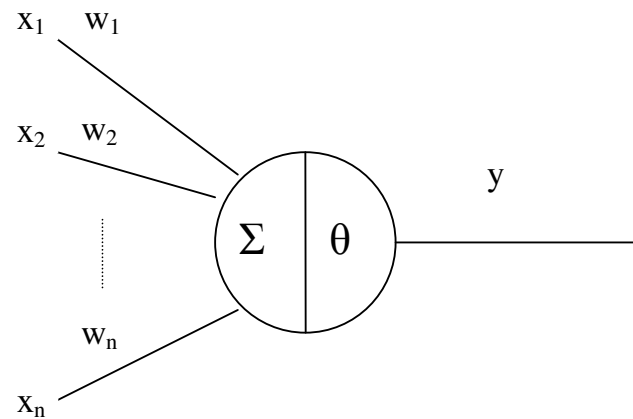


Figura 1.4. – Neurônio de McCulloch e Pitts. (BRAGA et al., 1998, p.9).

Os terminais de entrada do neurônio para emular o funcionamento das sinapses têm pesos (w_1, w_2, w_n) cujos valores podem ser positivos ou negativos (BRAGA et al., 1998, p.9). O efeito de cada uma das sinapses é calculado pelo produto $x_i w_i$. O corpo do neurônio artificial é constituído por um dispositivo que soma os diferentes produtos $x_i w_i$ (soma ponderada). Essa soma é comparada com o limiar θ (*threshold*) do neurônio: quando é superior ou igual a esse valor, o neurônio dispara. O neurônio de McCulloch e Pitts é ativado por meio de uma *função de ativação*, “que ativa a saída ou não, dependendo da soma ponderada das suas entradas”. A equação (1.1) descreve o funcionamento de um neurônio de McCulloch e Pitts.

$$\sum_{i=1}^n x_i w_i \geq \theta$$

(1.1)

Na equação (1.1) n é o número de entradas do neurônio e w_i é o peso associado a cada entrada x_i . O neurônio terá sua saída ativa somente quando o somatório dos produtos $x_i w_i$ for igual ou maior do que o limiar θ : essa característica do neurônio de McCulloch e Pitts é chamada de *propriedade tudo-ou-nada* (HAYKIN, 2001, p.39).

O neurônio proposto por McCulloch e Pitts tinha pesos fixos e os dois pesquisadores “simplificaram seu modelo assumindo que os nodos em cada camada da rede disparam sincronicamente, isto é, que todos os nodos em cada camada da rede são avaliados ao mesmo tempo e também as entradas em um instante de tempo t produzem a sua saída no tempo $t+1$.” (BRAGA et al., 1998, p.9).

A Figura 1.5 mostra um neurônio biológico (a) e um neurônio artificial (b), lado a lado, para melhor entender as analogias entre os dois (a letra θ indica o limiar do neurônio e a seta, a direção da transmissão do sinal):

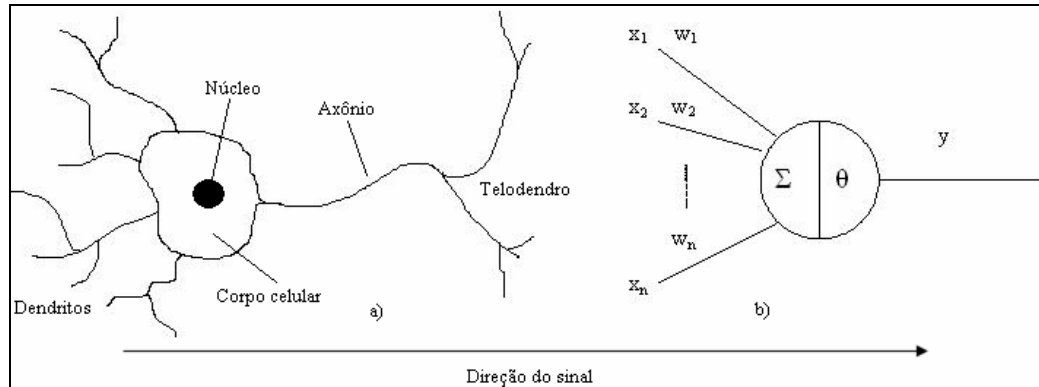


Figura 1.5. – (a) Neurônio biológico e (b) neurônio artificial.

O neurônio artificial de McCulloch e Pitts podia ter também um peso *bias*, cuja função é aumentar ou diminuir a entrada da função de saída do neurônio. Para mostrar os efeitos do *bias* sobre um neurônio artificial, utilizamos o esquema de neurônio artificial proposto por Haykin (Figura 1.6):

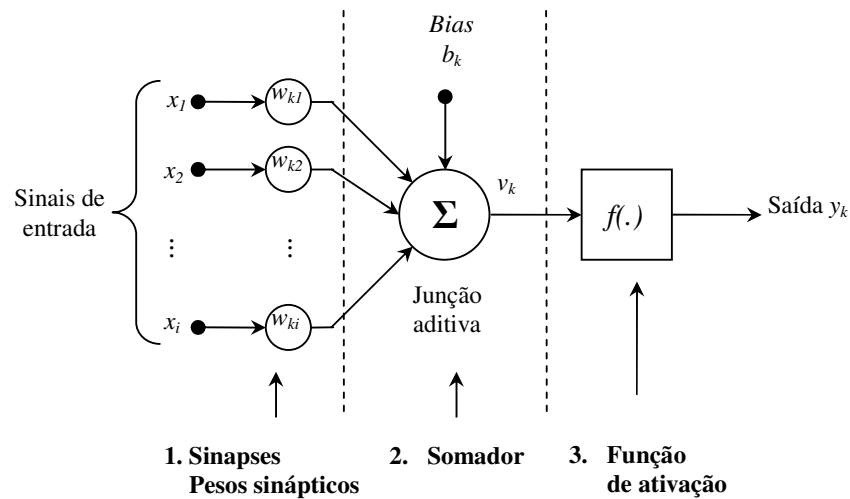


Figura 1.6. – Modelo não-linear de um neurônio (HAYKIN, 2001, p.36).

Para Haykin (2001, p.36), os três elementos básicos de um neurônio artificial são: um conjunto de *sinapses* ou *elos de conexão* (1), um *somador* (2) e uma *função de ativação* (3), chamada também de *função restritiva*, já que limita o intervalo do sinal de saída do neurônio.

O *bias* (indicado com b_k) tem como efeito aumentar ou diminuir a entrada da função de ativação. Em termos matemáticos, o funcionamento do neurônio é assim descrito por Haykin (2001, p.37):

$$u_k = \sum_{i=1}^n x_i w_{ki} \tag{1.2}$$

$$y_k = f(u_k + b_k) \tag{1.3}$$

$$v_k = (u_k + b_k) \tag{1.4}$$

$$y_k = f(v_k) \tag{1.5}$$

A equação (1.2) descreve a saída do somador do neurônio k . A segunda (1.3), representa a saída y_k do neurônio, e $f(.)$ é a função de ativação do neurônio. A soma de u_k com b_k (indicada com v_k) é chamada de *campo local induzido* ou *potencial de ativação do neurônio* (1.4). Caso o neurônio não tenha o *bias*, o potencial de ativação será igual à saída do somador. A última equação (1.5) é o resultado da substituição da (1.4) na (1.3).

O efeito do *bias* (negativo ou positivo) pode ser ilustrado pela Figura 1.7.:

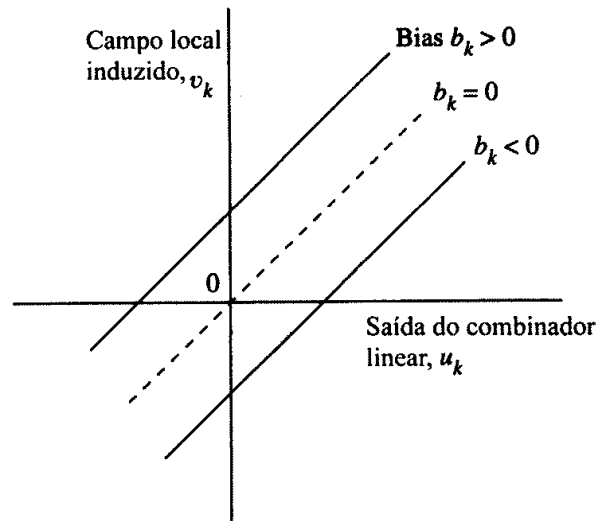


Figura 1.7. – Efeito do *bias* sobre o campo local induzido (HAYKIN, 2001, p.37).

Valores negativos do *bias* têm como efeito de reduzir o *potencial de ativação do neurônio* v_k , valores positivos de aumentá-lo. Quando a saída do somador u_k é igual a zero, o *potencial de ativação* é igual ao valor do *bias*.

Do ponto de vista funcional, o *bias* poderia ser considerado também como outra entrada x_0 , por exemplo, com seu peso relativo w_{k0} (se $x_0 = +1$, $w_{k0} = b_k$), como podemos ver na Figura 1.8.:

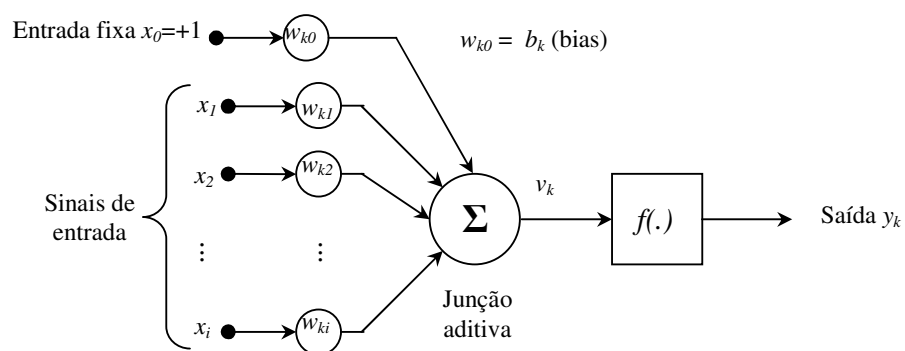


Figura 1.8. – Modelo de neurônio com *bias* como entrada fixa (HAYKIN, 2001, p.37)

Nesse caso, o potencial v_k de ação do neurônio é expresso pela equação (1.6):

$$v_k = \sum_{i=0}^n x_i w_{ki} \quad (1.6)$$

Depois de ter visto o funcionamento das sinapses, dos pesos, do somador e do *bias* (partes 1 e 2 do esquema proposto por Haykin, cf. p.27), analisaremos agora os principais tipos de funções de ativação.

1.5 Funções de ativação

Como vimos, a *função de ativação* é a que ativa a saída y_k do neurônio. Os principais tipos de funções de ativação utilizadas são quatro: a *função linear*, a *função rampa*, a *função de limiar* ou *degrau*, e a *função sigmoideal*.

A **função linear** (de ativação) é definida pela equação (1.7):

$$y = f(v_k) = av_k \quad (1.7)$$

Nessa equação a é um número real (inclinação da reta) que define a saída linear y para os valores de entrada v_k (potencial de ação do neurônio). A Figura 1.9. é a representação gráfica dessa função:

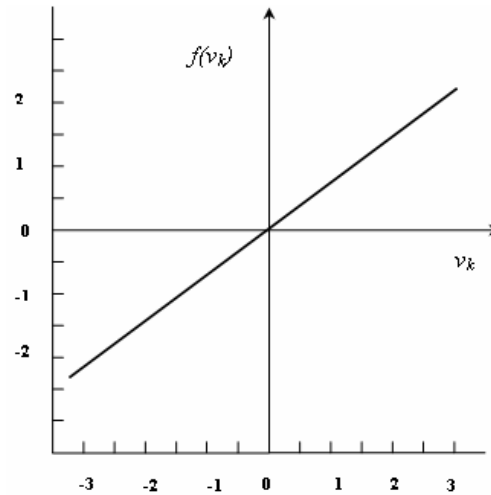


Figura 1.9. – Função Linear (BRAGA et al., 1998, p.10)

A função linear “pode ser restringida para produzir valores constantes em uma faixa $[-1, +1]$, e nesse caso, passa a ser a **função rampa**” (BRAGA et al., 1998, p.10), que pode ser representada pela eq. (1.8) e Figura 1.10:

$$f(v_k) = \begin{cases} 1 & \text{se } v_k \geq +1 \\ v_k & \text{se } +1 > v_k > -1 \\ -1 & \text{se } v_k \leq -1 \end{cases} \quad (1.8)$$

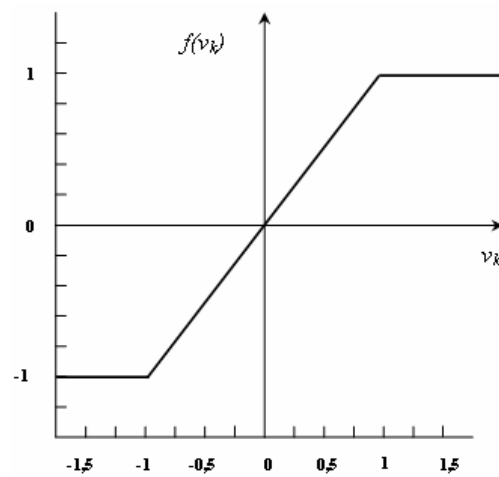


Figura 1.10. – Função Rampa

Esta função é chamada por Haykin de *função linear por partes* e “pode ser vista como uma aproximação de um amplificador não-linear” (HAYKIN, 2001, p.40).

A **função de limiar** (*threshold*) é uma das mais utilizadas na construção de redes neurais. Dois tipos são comumente utilizados:

$$\text{a) } f(v_k) = \begin{cases} 1 & \text{se } v_k \geq 0 \\ 0 & \text{se } v_k < 0 \end{cases} \quad (1.9)$$

$$\text{b) } f(v_k) = \begin{cases} 1 & \text{se } v_k \geq 0 \\ -1 & \text{se } v_k < 0 \end{cases} \quad (1.10)$$

O primeiro tipo de função de limiar (a) é conhecido também como **função de Heaviside** (HAYKIN, 2001, p.39) (Figura 1.11). Como vimos no começo do item 1.4, o neurônio de McCulloch e Pitts utiliza esse tipo de função. A saída de um neurônio assume o valor +1, se o potencial de ação v_k do neurônio é positivo, e 0 caso contrário (*propriedade tudo-ou-nada*) (HAYKIN, 2001, p.39). O segundo tipo de função de limiar (b) é conhecido também como **função sinal** (HAYKIN, 2001, p.41) (Figura 1.12):

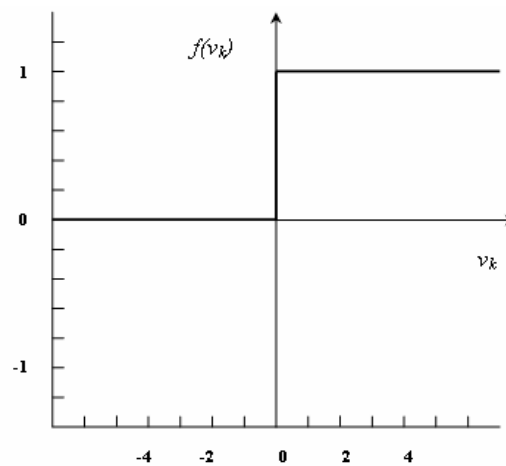


Figura 1.11. – Função de Limiar (a) (HAYKIN, 2001, p.41)

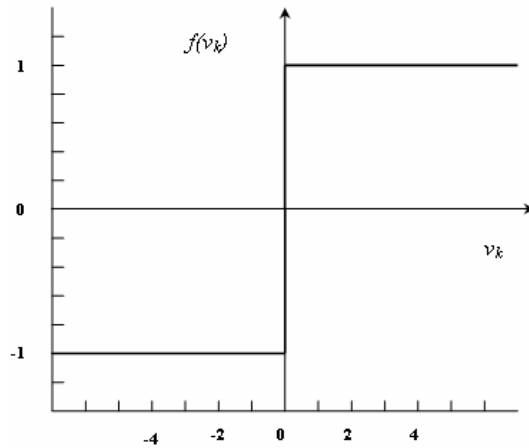


Figura 1.12 – Função de Limiar (b)

Segundo Haykin, a função mais utilizada no projeto de redes neurais é a **função sigmóide**, que “exibe um balanceamento adequado entre comportamento linear e não-linear” (HAYKIN, 2001, p.40). Outra característica importante dessa função é que, diferentemente da função de limiar, é contínua no intervalo de 0 a 1 (ou de -1 a 1) e diferenciável. Os dois tipos de função sigmóide mais utilizados são:

$$(a) f(v_k) = \frac{1}{1 + e^{-av_k}} \quad (1.11)$$

$$(b) f(v_k) = \frac{1 - e^{-av_k}}{1 + e^{-av_k}} \quad (1.12)$$

Haykin chama também o primeiro tipo (1.11) de **função logística** (HAYKIN, 2001, p.40). Os gráficos das duas funções são apresentados na Figura 1.13 e na Figura 1.14.

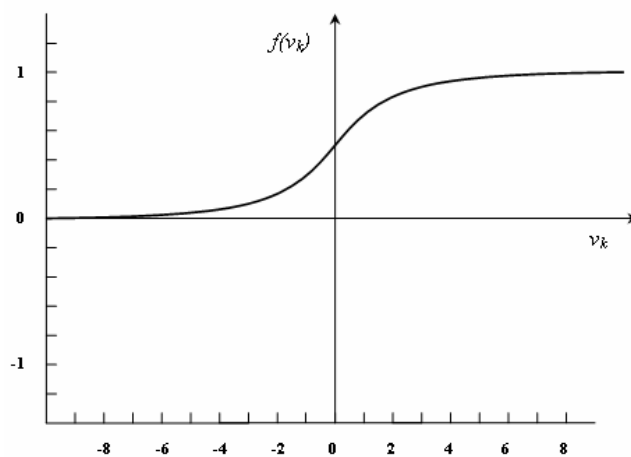


Figura 1.13 – Função sigmóide (a). Gráfico correspondente à Eq. (1.11). (HAYKIN, 2001, p.40), (BRAGA et al., 1998, p.10)

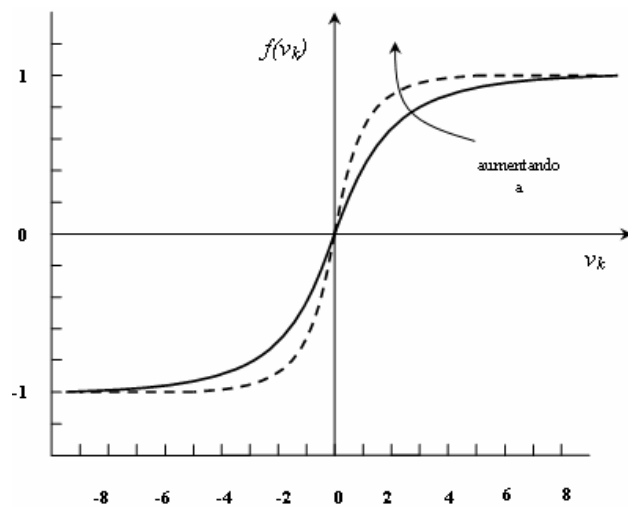


Figura 1.14. – Função sigmóide (b). Gráfico correspondente à Eq. (1.12).

O parâmetro a da função modifica a inclinação da função. Quando o valor do parâmetro tende a infinito, a função sigmóide se torna uma função de limiar.

1.6 Arquiteturas de RNAs

As redes neurais artificiais podem ser classificadas em função das seguintes características: número de camadas da rede, tipo de conexões entre os neurônios que as compõem e tipo de conectividade da rede.

Em uma rede neural, os diferentes neurônios são dispostos em camadas. No que diz respeito ao número de camadas as redes podem ser classificadas em *redes de camada única* e *redes de camadas múltiplas*.

A definição de rede de camada única leva em consideração somente a saída da rede, não sendo computada a camada de entrada, em cujos nós (nós fonte) não são realizadas operações computacionais (HAYKIN, 2001, p.47) (BRAGA et al., 1998, p.11). Braga et al. (BRAGA et al., 1998, p.11) dão a esse tipo de rede a seguinte definição: “rede de camada única – só existe um nó entre qualquer entrada e qualquer saída da rede”. Esquematicamente essa rede se apresentaria conforme mostrado na Figura 1.15:

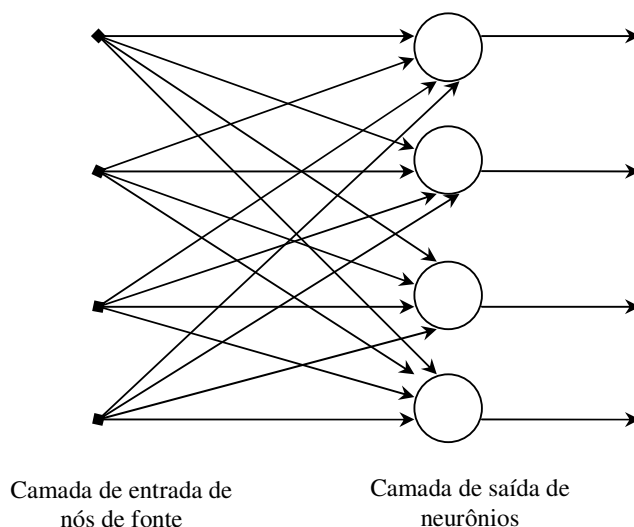


Figura 1.15. – Rede de camada única, *feedforward* (alimentada adiante), completamente conectada. (HAYKIN, 2001, p.47)

No exemplo da Figura 1.15, a rede conta com quatro nós de saída (neurônios).

As redes neurais de múltiplas camadas possuem uma ou mais camadas ocultas de neurônios, em outras palavras entre a entrada e a saída há mais de uma camada de neurônios.

A função das camadas ocultas é melhorar o desempenho da rede. A Figura 1.16. mostra um tipo de rede com várias camadas. Nesta rede, entre a entrada e a saída, temos uma camada de neurônios ocultos:

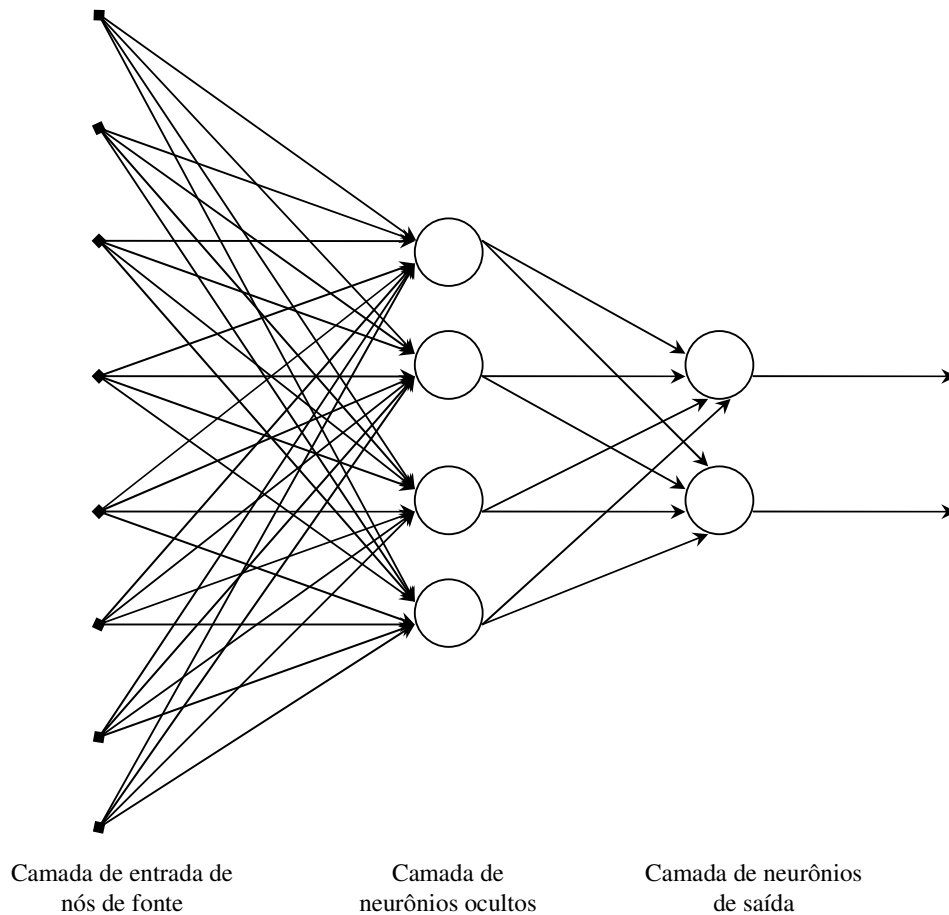


Figura 1.16. – Rede multicamadas, *feedforward*, completamente conectada (HAYKIN, 2001, p.48)

No que diz respeito ao tipo de conexões, as redes se dividem em *redes feedforward* (*acíclicas, alimentadas adiante, ou não-recorrentes*) e *redes feedback* (*cíclicas ou recorrentes*).

Nas redes *feedforward*, “a saída de um neurônio na i -ésima camada da rede não pode ser utilizada como entrada de nodos em camadas anteriores (índice igual ou menor que i)”

(BRAGA et al., 1998, p.11). A Figura 1.15, a Figura 1.16 e a Figura 1.18 ilustram esse tipo de rede.

Nas redes *feedback*, “a saída de algum neurônio na *i*-ésima camada da rede pode ser utilizada como entrada de nodos em camadas anteriores (índice igual ou menor que *i*)” (BRAGA et al., 1998, p.11). A Figura 1.17. ilustra esse tipo de rede.

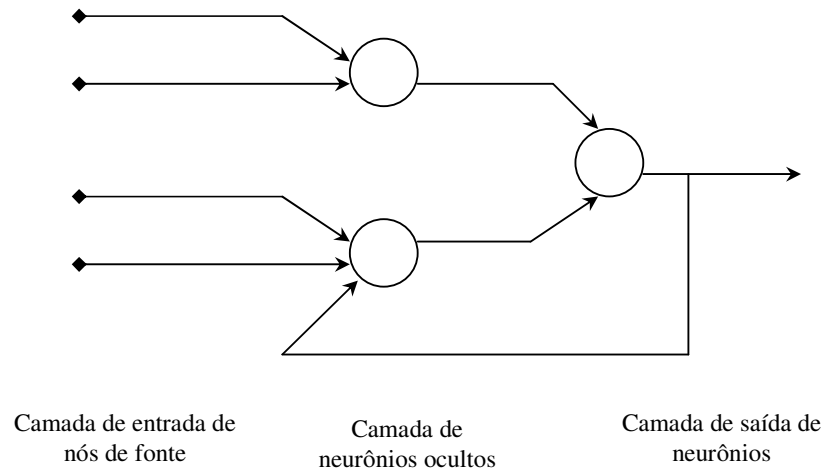


Figura 1.17. – Rede de múltiplas camadas, *feedback* e parcialmente conectada (BRAGA et al., 1998, p.12)

No que diz respeito à conectividade, as redes podem se dividir em *redes completamente conectadas* e *redes parcialmente conectadas*.

Uma rede é completamente conectada, quando todos os nós de uma camada estão conectados a todos os nós da camada seguinte. A Figura 1.15 e a Figura 1.16 ilustram esse tipo de rede. Quando isso não acontece, a rede é dita parcialmente ou fracamente conectada (BRAGA et al., 1998, p.12). A Figura 1.17. e a Figura 1.18 ilustram esse tipo de rede.

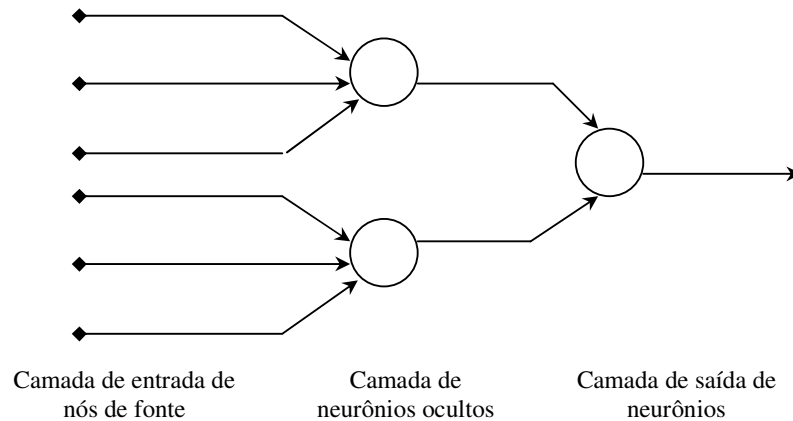


Figura 1.18 – Rede de múltiplas camadas, *feedforward* e parcialmente conectada (BRAGA et al., 1998, p.12)

Nesse capítulo apresentamos algumas das principais características das Redes Neurais Artificiais. No próximo examinaremos de modo mais aprofundado mais um tipo de Rede Neural, a rede de Kohonen.

2 Mapas auto-organizáveis: o modelo de Kohonen

Teuvo Kohonen, o finlandês inventor da rede homônima, é um dos pesquisadores de redes neurais artificiais que mais contribuiu para retomada dos estudos sobre a matéria. O objetivo de suas pesquisas foi, na década de 1980, o “de descobrir um modelo de auto-organização de informações em um processo de aprendizado indutivo capaz de ser usado como modelo para o aprendizado e organização de informações no neocórtex cerebral de um animal superior” (WANGENHEIM, 2006).

Esse modelo “deveria ser capaz de explicar como estímulos similares são aprendidos e agrupados em áreas próximas no cérebro de um animal e levam a uma posterior categorização desses estímulos e à formação subsequente de um modelo de mundo, criando uma explicação implícita do que foi percebido através dos agrupamentos de estímulos relacionados em categorias na memória e realizando o processo inteligente da abstração” (WANGENHEIM, 2006).

As pesquisas de Kohonen tiveram como ponto de partida o fato de que “conceitos similares parecem estar representados em áreas próximas no cérebro humano e que essa localização espacial deveria ser uma característica do aprendizado” (WANGENHEIM, 2006). Por essas razões, o modelo de rede neural artificial proposto por Kohonen talvez, seja o mais parecido com o funcionamento do cérebro humano.

A rede proposta por Kohonen é uma rede neural de mapeamento auto-organizável que utiliza um tipo de treinamento competitivo e não-supervisionado e que permite organizar os vetores (dados) de entrada em agrupamentos (*clusters*) (KOHONEN, 1989). Segundo Haykin, “o objetivo principal de um mapa-auto-organizável (SOM, *Self Organizing Map*) é transformar um padrão de sinal incidente de dimensão arbitrária em um mapa discreto uni- ou bidimensional e realizar esta transformação adaptativamente de uma maneira topologicamente ordenada” (HAYKIN, 2001, p.486).

Uma rede de Kohonen é geralmente composta por um reticulado bidimensional de neurônios e um conjunto de vetores pesos. Graficamente pode ser visualizada como mostrado na Figura 2.1:

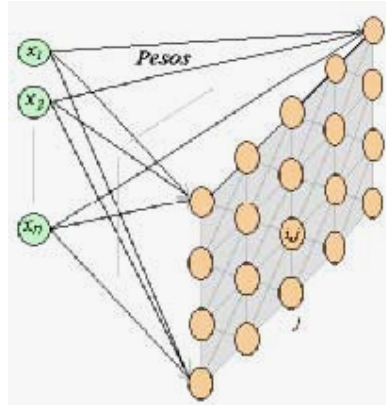


Figura 2.1. – Estrutura da rede neural de Kohonen. (MINUSSI, 2005)

Para entendermos o funcionamento desse tipo de rede começamos a definir alguns parâmetros. Seja m a dimensão do espaço de entrada:

$x = [x_1, x_2, \dots, x_m]^T$ o vetor de entrada.

O vetor peso sináptico de um neurônio j terá a mesma dimensão n do vetor de entrada:

$w_j = [w_{j1}, w_{j2}, \dots, w_{jm}]^T$, $j = 1, 2, \dots, n$. (n = número total de neurônios da rede)

A saída y_j do neurônio j , como mostrado na Figura 2.2, será igual:

$$y_j = \langle w_j, x \rangle$$

(2.1)

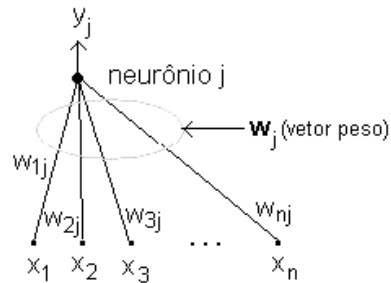


Figura 2.2. Conexão do vetor padrão de entrada X com o j -ésimo neurônio da rede de Kohonen. (MINUSSI, 2005)

A rede é inicializada atribuindo aos pesos sinápticos valores de 0 a 1 “tomados de um gerador de números aleatórios”, assim “nenhuma organização prévia é imposta ao mapa de características” (HAYKIN, 2001, p.487).

Depois da inicialização, há, segundo Haykin, três processos fundamentais que concorrem para a formação do mapa auto-organizável:

- 1) *Competição*. Para cada padrão de entrada, os neurônios da grade calculam seus respectivos valores de uma função discriminante. Essa função discriminante fornece a base para a competição entre os neurônios. O neurônio particular com o maior valor da função discriminante é declarado vencedor da competição. (HAYKIN, 2001, p.487)
- 2) *Cooperação*. O neurônio vencedor determina a localização espacial de uma vizinhança topológica de neurônios excitados, fornecendo assim a base para a cooperação entre os neurônios vizinhos. (HAYKIN, 2001, p.487)
- 3) *Adaptação Sináptica*. Este último mecanismo permite que os neurônios excitados aumentem seus valores individuais da função discriminante em relação ao padrão de entrada através de ajustes adequados aplicados a seus pesos sinápticos. Os ajustes feitos são tais que a resposta do neurônio vencedor à aplicação subsequente de um padrão de entrada similar é melhorada. (HAYKIN, 2001, p.487)

O processo competitivo permite identificar o *neurônio vencedor* ou *neurônio melhor casado*. O neurônio vencedor terá, em seguida, seus pesos atualizados.

O neurônio vencedor é aquele que, na rede, apresenta a maior atividade (y). Para encontrar o neurônio vencedor podemos proceder de duas maneiras diferentes:

- a) Podemos comparar todos os produtos internos $\mathbf{w}_j^T \mathbf{x}$ ($j = 1, 2, \dots, m$). O neurônio vencedor (NV) será aquele cuja saída (produto interno) tiver o valor mais alto:

$$NV = \max \mathbf{w}_j^T \mathbf{x}, \quad j = 1, 2, \dots, n \quad (2.2)$$

Ou

b) utilizar o critério da distância euclidiana. O neurônio vencedor (melhor casado) será aquele cujo vetor de pesos sinápticos \mathbf{w}_j tiver a menor distância em relação ao vetor de entrada \mathbf{x} :

$$NV = \min \|\mathbf{x} - \mathbf{w}_j\|, \quad j = 1, 2, \dots, n \quad (2.3)$$

Do ponto de vista gráfico, esse processo pode ser representado pelo esquema da Figura 2.3, onde o vetor de pesos w_s do neurônio vencedor S é o que mais se aproxima do vetor de pesos do vetor de entrada x , e Δw_s é o erro de aproximação entre os dois vetores.

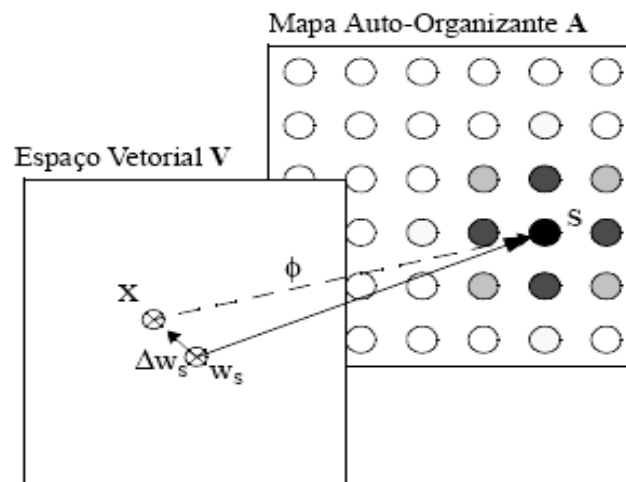


Figura 2.3.– Mapeamento de vetores em uma rede de Kohonen. (WANGENHEIM, 2006)

O processo competitivo de uma rede de Kohonen, por meio do qual se determina o neurônio vencedor, é comparável ao modelo de espaço vetorial (cf. § 4.2). No espaço vetorial também a comparação é feita por meio da distância euclidiana entre vetores (medida do cosseno): valores mais próximos de 1 indicam que o vetor de busca e o vetor de pesos do

documento são parecidos, valores mais próximos de zero indicam que os dois vetores são diferentes.

O que diferencia substancialmente uma rede de Kohonen do modelo de espaço vetorial são os dois processos sucessivos, ou seja, os processos cooperativo e adaptativo.

Em uma rede de Kohonen, o neurônio vencedor atualiza também os pesos sinápticos dos neurônios vizinhos, em particular dos mais próximos, aumentando seus valores. Esse processo é chamado por Haykin *cooperativo* (HAYKIN, 2001, p.489). A função utilizada por Kohonen para modificar os pesos, que é chamada de *função de vizinhança*, é uma função gaussiana, cuja equação (relativa a uma grade bidimensional) é a que segue:

$$h_{j,i(x)} = \exp \left(- \left(\frac{\|r_j - r_i\|}{2\sigma^2} \right)^2 \right) \quad (2.4)$$

Na equação (2.4), r_i é a posição discreta do neurônio vencedor i e r_j é a posição do neurônio excitado j , neurônio cujo vetor também será modificado pela rede. O parâmetro σ representa a largura efetiva da vizinhança topológica (HAYKIN, 2001, p.489). Além disso, no algoritmo da rede de Kohonen, a largura de σ diminui no tempo. Uma das escolhas mais utilizadas para o decaimento de σ no tempo é o decaimento exponencial, que é representado pela seguinte equação:

$$\sigma = \sigma_0 \exp \left(- \frac{n}{\tau_1} \right) \quad n = 0, 1, 2, \dots \quad (2.5)$$

Na equação (2.5), σ_0 é o valor de σ quando a rede é inicializada e τ_1 é uma constante de tempo.

Substituindo a equação (2.5) na (2.4), obtemos, para a modificação da vizinhança topológica, a seguinte fórmula:

$$h_{j,i(x)} = \exp - \left(\frac{\|r_j - r_i\|^2}{2\sigma^2(n)} \right) \quad n = 0, 1, 2, \dots \quad (2.6)$$

Utilizando essa equação quando o “tempo n (i.e., o número de interações) aumenta, a largura $\sigma(n)$ decresce a uma taxa exponencial e a vizinhança topológica diminui de uma maneira correspondente” (HAYKIN, 2001, p.490).

A Figura 2.4 ilustra graficamente os efeitos da função de vizinhança sobre os neurônios:

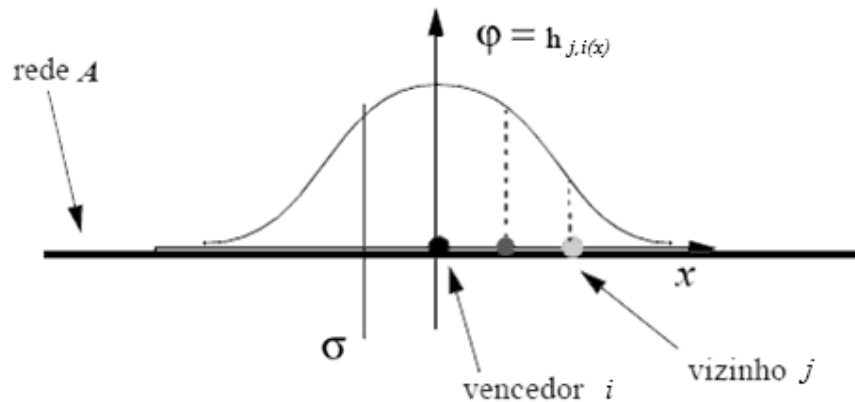


Figura 2.4. – Forma da função de vizinhança. (WANGENHEIM, 2006)

Segundo Haykin, por *processo adaptativo sináptico* entende-se a capacidade que o “vetor w_j do neurônio j se modifique em relação ao vetor de entrada x ” (HAYKIN, 2001, p.491). A adaptação dos pesos é feita segundo as equações (2.7) e (2.8):

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n) (x(n) - w_j(n)) \quad | \quad (\forall w_j(n+1) \in \text{a vizinhança}) \quad (2.7)$$

$$w_j(n+1) = w_j(n) \quad | \quad (\forall w_j(n+1) \notin \text{a vizinhança}) \quad (2.8)$$

Onde $\mathbf{w}_j(n+1)$ é o novo vetor de pesos, $\eta(n)$ é a taxa de aprendizagem, $h_{j,i(x)}(n)$ é a função de vizinhança e $\mathbf{x}(n)$ o vetor de entrada. A taxa de aprendizagem, a partir de um valor inicial η_0 , deve decrescer gradualmente em função do tempo.

Aqui terminamos a apresentação do modelo de rede proposto por Kohonen. No que diz respeito à representação gráfica desse tipo de rede, veremos mais detalhes no Capítulo 6, quando implementaremos nossa rede de Kohonen.

3 Reconhecimento de padrões e aprendizado

Neste capítulo, serão abordadas algumas questões genéricas ligadas ao reconhecimento de padrões e alguns conceitos-chave de Inteligência Artificial (IA). Em seguida, faremos um breve histórico do aprendizado na IA e exporemos uma classificação dos métodos de aprendizado. No que diz respeito ao aprendizado, analisaremos o modo de apresentar os padrões a serem aprendidos, o processo de aprendizado, a maneira de avaliá-lo e os requisitos necessários para o reconhecimento de padrões.

3.1 Conceitos-chave

O que são padrões?

Padrões são unidades de informação que se repetem, ou então, são seqüências de informações que dispõem de uma estrutura que se repete.

O que é o reconhecimento de padrões?

O reconhecimento de padrões (RP) é a transformação da informação do nível subsimbólico (sinais), ao nível simbólico (o do conhecimento). Mais genericamente, o reconhecimento de padrões é um ramo da Ciência que trata da classificação e da descrição de objetos.

Reconhecer padrões é uma das atividades mais recorrentes e mais importantes do ser humano, que a pratica constantemente e que lhe permitiu uma relativa rapidez na evolução, em comparação com os demais seres vivos que povoam o planeta Terra.

Mais especificamente, o que é reconhecer padrões? Lendo um texto escrito, os olhos capturam os sinais representados pelos grafemas (letras) da língua, transformam-nos em sinais nervosos, que são reconhecidos (interpretados) pelo cérebro humano. Este último reconhece aqueles signos, palavras, frases e outros elementos, em um crescendo de informações, até chegar ao sentido do texto, ou seja, até entender o texto (RAUBER, 1997).

Os homens utilizam constantemente suas habilidades em reconhecer padrões, essencialmente por meio dos sentidos. Assim pode reconhecer rostos, músicas, vinhos, materiais etc.

Segundo Rauber (RAUBER, 1997), “reconhecimento de padrões é a transformação da informação do nível subsimbólico (sinais), ao nível simbólico (o do conhecimento)”¹.

Mais genericamente, poderíamos dizer que o reconhecimento de padrões (RP) é um ramo da Ciência que trata da classificação e da descrição de objetos (MARQUES de SÁ, 2006).

A Figura 3.1 representa de modo esquemático as etapas necessárias para a realização de um projeto que envolva o reconhecimento de padrões:



Figura 3.1. – Etapas de um projeto sobre reconhecimento de padrões (MARQUES de SÁ, 2006)

¹ *Pattern recognition is transforming information from the subsymbolic level (signals) to the symbolic level (meanings).*

De acordo com a Figura 3.1, após a definição dos objetos a serem analisados, é necessária a implementação de um módulo que consiga extrair as características dos objetos a serem classificados (ou a serem descritos) (*Extrator de características*). Em seguida, é necessário realizar uma seleção mais discriminativa das características extraídas (*Seletores de características*). Por último, vem a implementação do *Classificador*, que conclui a fase de reconhecimento de padrões.

São inúmeros os setores de aplicação de modelos (sistemas) de reconhecimento de padrões, por isso a área de estudo sobre RP tem caráter interdisciplinar e transdisciplinar. Dentre as áreas em que mais é utilizado o reconhecimento de padrões, estão:

Aprendizado de máquina (*machine learning*)

Conjuntos nebulosos (*fuzzy sets*)

Inteligência artificial

Linguagens formais

Mineração de dados (*data mining, text mining and knowledge discovery*)

Modelagem estrutural

Modelagem neuronal (*artificial neural networks*)

Processamento de sinal e imagem

Sistemas adaptativos

Teoria da otimização e da estimação

Teoria dos autômatos.

A área que nos concerne mais de perto é a das Redes Neurais Artificiais, área sobre a qual trabalharemos nos próximos itens.

3.2 Inteligência Artificial “Simbólica” e “Subsimbólica” e aprendizado: definições preliminares

Segundo von Wangenheim, podemos dividir a Inteligência Artificial “em **IA Simbólica** (ou Tradicional) e **IA Subsimbólica** (ou Conexionista). As fronteiras entre essas duas categorias de IA, porém, não são muito claras” (WANGENHEIM, 2006).

Ainda de acordo com esse autor, a IA Simbólica trabalha com “peças de informação onde cada elemento de informação possui um significado a ele associado (ou, no mínimo, associável)” (WANGENHEIM, 2006), ou seja, cada dado possui um simbolismo. A IA Simbólica trabalha “com procedimentos lógicos e algorítmicos e um dos objetivos é o desenvolvimento de lógicas e de metodologias de representação do conhecimento de forma a possibilitar uma melhor manipulação dessas representações simbólicas” (WANGENHEIM, 2006).

Na IA Subsimbólica se trabalha, segundo von Wangenheim, “com dados ‘numéricos’, onde nem sempre é possível associar-se um simbolismo claro a um dado específico no seu contexto de aplicação” (WANGENHEIM, 2006) e se utilizam “métodos numéricos para manipulação de grandes quantidades de dados (geralmente representados como padrões) com o objetivo de abstrair suas características subsimbólicas e gerar classificações ou raciocínios sobre esses dados” (WANGENHEIM, 2006). As Redes Neurais Artificiais são um claro exemplo de IA Subsimbólica.

Sempre de acordo com von Wangenheim, “algumas técnicas da IA podem ser classificadas tanto como simbólicas como subsimbólicas, dependendo de como a informação é encarada e codificada”. O autor indica, como exemplo disso, as redes de Bayes e a Lógica *Fuzzy*. As primeiras podem ser utilizadas tanto para classificar padrões (subsimbólicos) como para classificar estruturas de conhecimento (WANGENHEIM, 2006). Este autor afirma que, “da mesma forma, uma função de pertinência em lógica difusa [nebulosa] pode possuir um significado explícito ou pode ser simplesmente o resultado de um algoritmo numérico aplicado a um conjunto de dados” (WANGENHEIM, 2006).

Segundo von Wangenheim, “a capacidade de um sistema de realizar o Reconhecimento de Padrões de forma flexível e adaptável está intimamente associada à idéia de que **um sistema de reconhecimento de padrões deve ser capaz de aprender as características e a distribuição dos padrões** no espaço vetorial definido por um determinado domínio de aplicação” (WANGENHEIM, 2006). Isso implica que o sistema seja

conseqüentemente capaz de “aprender como **associar** um determinado padrão à **classe** à qual pertence” (WANGENHEIM, 2006).

Von Wangenheim (2006) indica quatro fases distintas na evolução dos estudos sobre o aprendizado em IA:

- 1) A Fase Subsimbólica (1950 - 1965)
- 2) A Fase Simbólica (1962 - 1975)
- 3) Sistemas de Aprendizado baseados em Conhecimento (1976 - 1988)
- 4) Sistemas Híbridos e Integrados (1988 - ...)

O começo da **Fase Subsimbólica** é ligado aos estudos de McCulloch e Pitts sobre o neurônio artificial e aos estudos dos demais pesquisadores sobre as redes neurais. Os trabalhos de Minsky e Papert (cf. p.20) marcaram o fim dessa fase nas pesquisas sobre o aprendizado.

Para von Wangenheim, “a grande contribuição desta primeira fase das pesquisas na área do aprendizado foi o surgimento do termo **Reconhecimento de Padrões** e a definição de sua área de atuação” (WANGENHEIM, 2006).

A **Fase Simbólica** começa no início da década de 1960. Nesse período, as pesquisas sobre o aprendizado em IA “passaram a ser influenciadas por modelos cognitivos do processo de aprendizado humano” (WANGENHEIM, 2006), utilizando métodos e conceitos da **Psicologia Cognitiva**.

Segundo von Wangenheim “o termo ‘simbólico’ desta fase do desenvolvimento de algoritmos de aprendizado, que está intimamente relacionado com o termo ‘explícito’ em IA, significa que a interpretação dos dados ou padrões a serem aprendidos depende unicamente de informações explicitamente representadas no sistema e de regras ou conhecimento explicitamente descritos, não havendo representações de caixa-preta (como nos *Perceptrons*) ou representações implícitas sob a forma de algum algoritmo numérico complexo” (WANGENHEIM, 2006).

Por essas razões, o aprendizado simbólico de conceitos utilizava “ao invés de métodos numéricos ou estatísticos, representações lógicas ou estruturadas (como grafos) de informações” (WANGENHEIM, 2006).

A terceira fase, a dos **Sistemas de Aprendizado baseados em Conhecimento**, se distingue da precedente pela busca de “metodologias generalizadas para o aprendizado” (WANGENHEIM, 2006). Nessa fase se estabeleceram “os dois principais paradigmas do

aprendizado de máquina, o **Aprendizado Baseado em Instâncias** e o **Aprendizado Baseado em Modelos**” (WANGENHEIM, 2006).

De acordo com von Wangenheim (2006):

um enfoque da pesquisa se concentra em grandes sistemas que são capazes de incorporar uma grande quantidade de conhecimento sobre um determinado domínio de aplicação e de realizar raciocínios complexos nesse domínio, assim como aprender novos fatos analisando situações. Esta área originou tanto o Raciocínio como o Aprendizado Baseado em Modelos (*Model-Based Reasoning* e *Model-Based Learning*).

A quarta e atual fase é a dos **Sistemas Híbridos e Integrados**, que se caracteriza por uma “integração de técnicas e paradigmas das diversas fases anteriores” (WANGENHEIM, 2006). Nessa última fase houve uma forte retomada dos estudos sobre redes neurais artificiais.

3.3 Métodos de aprendizado e classificação de padrões

O Aprendizado de Máquina (*Machine Learning*) é uma subárea (ramo) da Inteligência Artificial que desenvolve métodos que permitem a computadores aprenderem e melhorarem seus desempenhos.

Do ponto de vista do reconhecimento de padrões, o aprendizado de máquina é, segundo von Wangenheim (2006), “um método que nos permite organizar uma seqüência de exemplos de padrões P_1, \dots, P_n em vários conjuntos de padrões CP_1, \dots, CP_k , denominados **classes**, de tal forma que os padrões organizados em cada conjunto são **similares entre si e dissimilares dos padrões armazenados nos outros conjuntos.**” Esse processo de aprendizado é sintetizado na Figura 3.2:

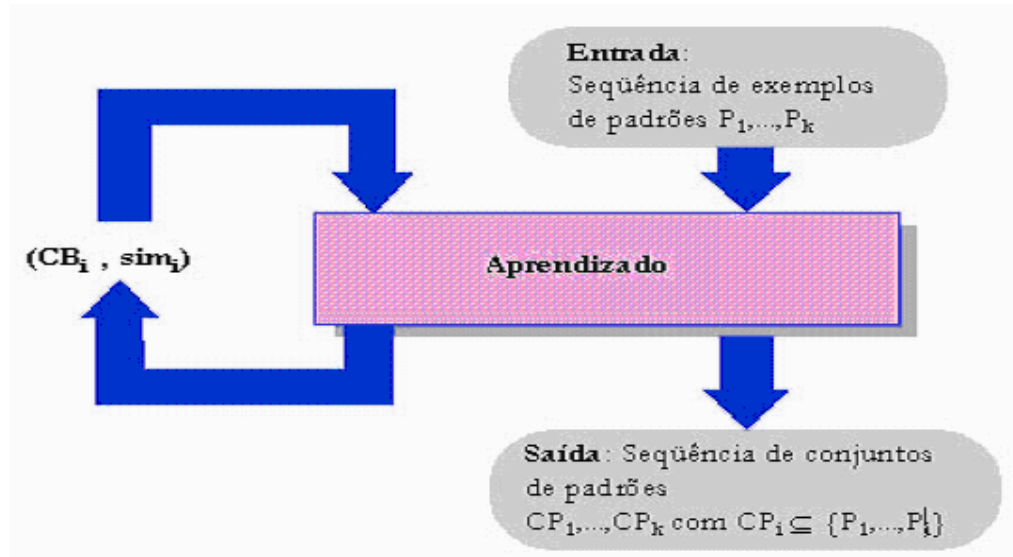


Figura 3.2.– Aprendizado de máquina (WANGENHEIM, 2006)

Segundo von Wangenheim (2006), os métodos de aprendizado podem ser classificados de acordo com três eixos:

1. forma de apresentação dos padrões a serem aprendidos
2. processo de aprendizado
3. forma de avaliação do aprendizado.

1) No que diz respeito à forma de apresentação dos padrões a serem aprendidos, de modo geral, o aprendizado pode ser dividido em **aprendizado supervisionado** e **aprendizado não-supervisionado**. No aprendizado supervisionado, “existe a figura do professor, que apresenta os exemplos a serem aprendidos e controla a avaliação da qualidade do aprendizado realizado, avaliando se um determinado conjunto de padrões de treinamento foi aprendido ou não” (WANGENHEIM, 2006).

Em RP, existem duas categorias de problemas: os **problemas classificatórios** e os **problemas exploratórios** (WANGENHEIM, 2006).

— Os problemas classificatórios são problemas em que as classes de padrões são previamente conhecidas. “O objetivo do aprendizado aqui é criar um classificador capaz de replicar este conhecimento e, eventualmente, refiná-lo” (WANGENHEIM, 2006).

— Os problemas exploratórios são problemas em que temos como ponto de partida padrões não previamente classificados. O objetivo do aprendizado é justamente o de agrupar esses padrões em classes.

Em problemas de tipo classificatório são utilizados métodos de aprendizado supervisionado, em quanto que, para problemas de tipo exploratório, são utilizados métodos de aprendizado não-supervisionado. Podemos visualizar esses processos de modo esquemático na Figura 3.3.

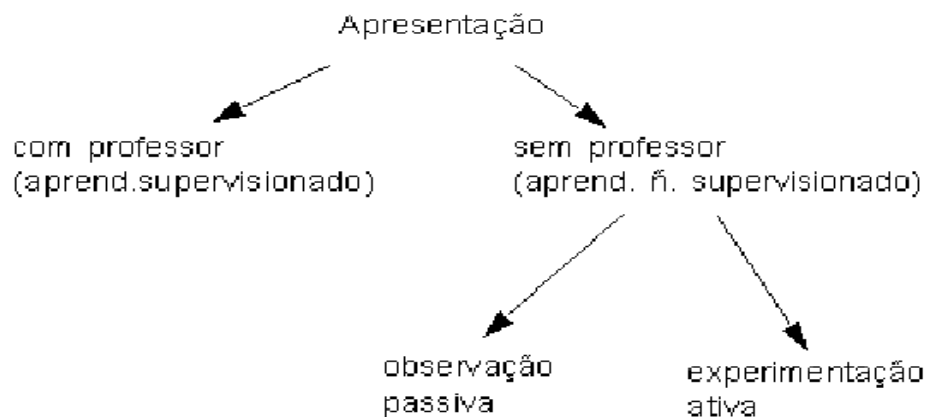


Figura 3.3.– Forma de apresentação dos padrões a serem aprendidos: aprendizado supervisionado e aprendizado não-supervisionado (WANGENHEIM, 2006)

Segundo von Wangenheim, o aprendizado supervisionado pode ser ulteriormente subdividido em duas categorias: **observação passiva** e **experimentação ativa** (WANGENHEIM, 2006). A observação passiva indica que o sistema se limita à simples análise dos dados. A experimentação ativa indica que “o sistema pode construir hipóteses com base nos dados e testar suas conclusões” (WANGENHEIM, 2006).

2) Os resultados de todo processo de aprendizado são novos conhecimentos, que se apresentam sob forma de novas conclusões. Segundo von Wangenheim (WANGENHEIM, 2006), “do ponto de vista lógico, uma conclusão **C** pode ser sempre representada da seguinte forma:

$$P_1, \dots, P_n \rightarrow C$$

onde P_1, \dots, P_n são o conjunto de **premissas** que justificam a conclusão”.

As conclusões de um processo de aprendizado podem ser *conclusões lógicas*, *abstrações* ou *analogias* (WANGENHEIM, 2006). O tipo de conhecimento gerado depende, segundo von Wangenheim, “do **modelo de inferência** subjacente ao processo” (WANGENHEIM, 2006), que “determina também a **semântica** de uma conclusão à qual um sistema de aprendizado chegou”. A Figura 3.4 representa esquematicamente os diferentes tipos de modelos de inferência:

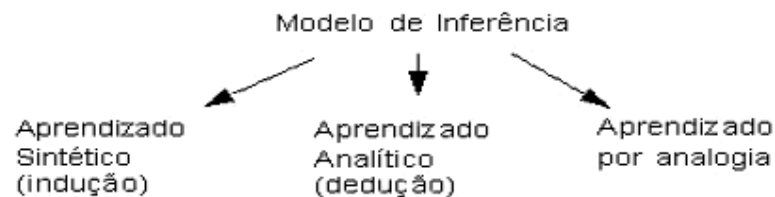


Figura 3.4. – Modelos de inferência utilizados no aprendizado (WANGENHEIM, 2006)

Os três modelos de aprendizado que podem ser utilizados são:

- a) o aprendizado analítico, realizado por **dedução**,
- b) o aprendizado sintético, realizado por **indução** e
- c) o aprendizado por **analogia** (WANGENHEIM, 2006).

— “No **aprendizado analítico** são realizados **raciocínios dedutivos**” (WANGENHEIM, 2006). Para que uma conclusão seja verdadeira é necessário que todas as premissas também sejam verdadeiras. A dedução é pouco utilizada em RP.

— “No **aprendizado sintético** são realizados **raciocínios indutivos**” (WANGENHEIM, 2006). As conclusões são induzidas a partir de fatos concretos. O método indutivo é o caminho privilegiado utilizado na ciência para descobrir novos fenômenos e o mais utilizado para reconhecimento de padrões.

— No **aprendizado por analogia**, as conclusões são determinadas por analogia. Modelos, regras e premissas existentes são aplicadas a novos fatos, podendo ou não gerar conclusões

verdadeiras. O aprendizado por analogia é raramente utilizado em RP (WANGENHEIM, 2006).

3) A avaliação do aprendizado é feita em função dos resultados obtidos: quando o objetivo é alcançado plenamente ou de maneira suficiente, o processo de aprendizado termina. Segundo von Wangenheim, “a determinação do objetivo de aprendizado pode ocorrer de diferentes formas”(WANGENHEIM, 2006). A Figura 3.5 ilustra graficamente esse processo:

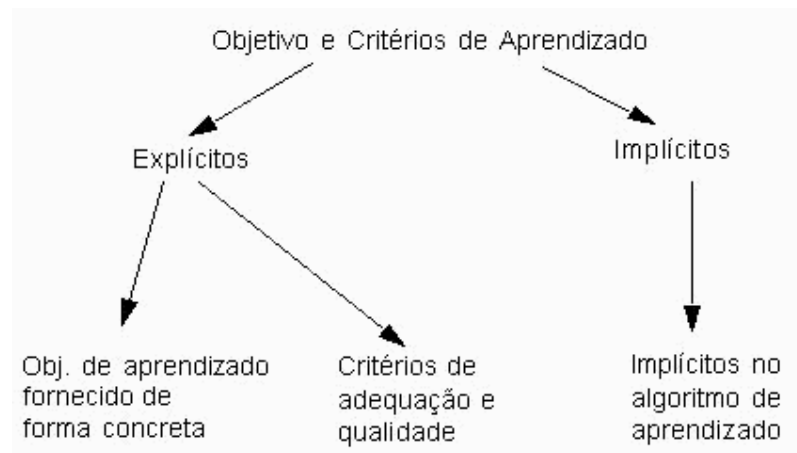


Figura 3.5. – Objetivo e critérios de aprendizado (WANGENHEIM, 2006)

Um objetivo pode ser definido de forma explícita quando são dadas aos sistemas as classes de padrões que queremos que o sistema aprenda (WANGENHEIM, 2006). Podem ser definidos critérios de qualidade quando é definida a taxa de erros com a qual devem ser aprendidas as classes fornecidas ao sistema, e quando é desconhecido o número de classes que “existem em nosso conjunto de padrões, mas definimos a taxa de erros mínima aceitável para uma classificação a ser aprendida” (WANGENHEIM, 2006).

Enfim, na avaliação do aprendizado podem ser também utilizados “algoritmos iterativos e possuir um critério de avaliação implícito (que às vezes nada mais é do que um dos citados anteriormente)” (WANGENHEIM, 2006).

4 A transformação de textos em vetores

Depois de ter passado por um ou mais de um dos processos vistos no capítulo precedente, os textos podem ser analisados ou tratados por outros programas para diversos fins. Tendo em vista que nosso objetivo é trabalhar com redes neurais, deveremos estudar as técnicas que permitem transformar textos/unidades lexicais em vetores/padrões. Para isso analisaremos dois modelos clássicos utilizados em Recuperação de Informações (*Information Retrieval*): o **Modelo booleano** e o **Modelo de espaço vetorial**.

O texto de referência sobre Recuperação de Informações é o livro de Ricardo Baeza-Yates e Berthier Ribeiro-Neto, **Modern Information Retrieval** (1999).

Os modelos clássicos em Recuperação de Informações consideram que cada documento d_j é descrito do ponto de vista lógico (*document logical view*) por um conjunto de palavras chaves (*keywords*) representativas chamadas termos índices (*index term*) k_i (BAEZA-YATES; RIBEIRO-NETO, 1999, 24). Em geral, são escolhidos como termos índices nomes; adjetivos, advérbios e conetivos em geral são descartados. Cabe lembrar a eliminação de muitas palavras, por exemplo das *stopwords*, é feita nas fases de pré-processamento de um documento. No entanto é possível também considerar como termos índices todas as palavras contidas em um documento, nesse caso, a vista lógica do documento é *full text* (BAEZA-YATES; RIBEIRO-NETO, 1999, p.24).

Uma vez fixado um conjunto de termos índices por um documento, para atribuir uma importância relativa a certo termo k_i em um documento d_j , é atribuído um peso a esse termo w_{ij} , relativo ao par (k_i, d_j) .

4.1 Modelo Booleano

O Modelo booleano é um dos modelos de Recuperação de Informações mais simples e se baseia na álgebra booleana. Por seu formalismo elegante e relativamente simples, por suas consultas, feitas por meio de expressões booleanas, de semântica precisa, esse modelo teve grande sucesso no passado (BAEZA-YATES; RIBEIRO-NETO, 1999, p.26).

Vejam agora, na **Tabela 4.1**, os conceitos básicos utilizados para descrever o modelo booleano:

Tabela 4.1 Conceitos básicos e símbolos utilizados para descrever o modelo booleano. (KAESTNER, 2006), (BAEZA-YATES; RIBEIRO-NETO, 1999, p.25)	
–	k_i : termo índice;
–	d_j : documento,
–	N : número total de documentos;
–	$K = (k_1, k_2, \dots, k_t)$ conjunto de termos índices, onde
–	t : número total de termos índices na coleção de documentos;
–	$w_{ij} \geq 0$: peso associado a (k_i, d_j) ;
–	$w_{ij} = 0$: indica que o termo índice não pertence ao documento;
–	$\vec{d}_j = (w_{1j}, w_{2j}, \dots, w_{tj})$ é o vetor formado pelos pesos associados ao documento d_j ;
–	g_i : é a função que retorna o peso associado ao termo índice k_i , i.e., $g_i(\vec{d}_j) = w_{ij}$

O Modelo booleano considera que os termos índices ($K=\{k_1, k_2, \dots, k_t\}$) estejam presentes ou ausentes em um determinado documento (d_j). Por esta razão o peso de cada termo tem valor binário ($w_{i,j} \in \{0, 1\}$) (BAEZA-YATES; RIBEIRO-NETO, 1999, p.26).

Uma consulta q é composta de termos índices k_i ligados por três operadores (*not*, *and*, *or*). Assim uma consulta é essencialmente uma expressão booleana que pode ser representada como uma disjunção de vetores conjuntivos, em outras palavras em *disjunctive normal form* - DNF (BAEZA-YATES; RIBEIRO-NETO, 1999, p.26) (F.N.D. Forma normal disjuntiva).

- Considerando:

$$q = k_a \wedge (k_b \vee \neg k_c)$$

escrito na forma normal disjuntiva:

$$\bar{q}_{fnd} [\text{na f.n.d.}] = (k_a \wedge k_b \wedge k_c) \vee (k_a \wedge k_b \wedge \neg k_c) \vee (k_a \wedge \neg k_b \wedge \neg k_c)$$

$$\bar{q}_{fnd} [\text{na f.n.d.}] = \{k_a, k_b, k_c\} \vee \{k_a, k_b\} \vee \{k_a\}$$

$$\bar{q}_{fnd} [\text{na f.n.d.}] = (1,1,1) \vee (1,1,0) \vee (1,0,0)$$

Nessa forma, cada componente de q (e.g. $\bar{q}_{cc}=(1,1,0)$) é um vetor de pesos binários associado com a tupla (k_a, k_b, k_c) , que é chamado de **componente conjuntivo** (\bar{q}_{cc}) ou **conjunto** de \bar{q}_{fnd} (BAEZA-YATES; RIBEIRO-NETO, 1999, p.26).

A Figura 4.1 representa graficamente os três componentes conjuntivos da consulta $[q = k_a \wedge (k_b \vee \neg k_c)]$ [lembrando que $\bar{q}_{fnd} = (1,1,1) \vee (1,1,0) \vee (1,0,0)$]:

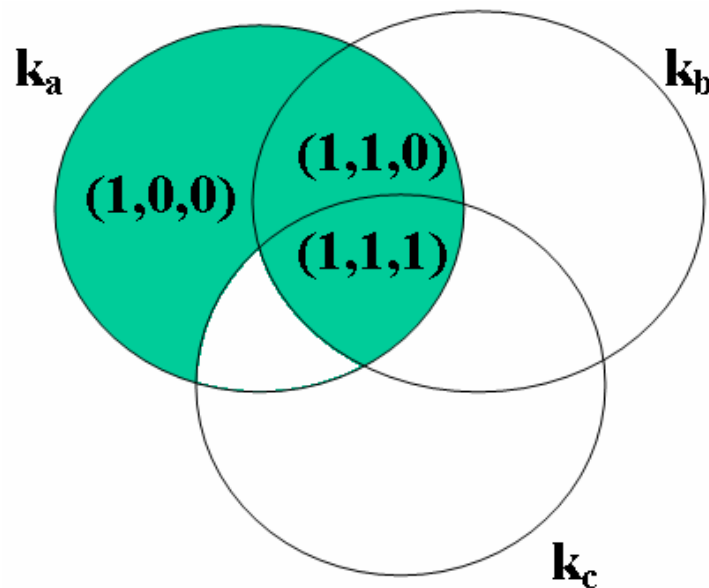


Figura 4.1. – Representação gráfica dos três componentes conjuntivos da consulta $[q = k_a \wedge (k_b \vee \neg k_c)]$. (KAESTNER, 2006)

Cada componente conjuntivo \bar{q}_{cc} representa um conjunto de documentos ideais. A consulta q é satisfeita por um documento d_j quando este documento d_j é um dos documentos ideais do conjunto.

No modelo booleano, a similaridade entre um documento d_j e uma consulta q é assim definida (BAEZA-YATES; RIBEIRO-NETO, 1999, p.26):

$$sim(q, d_j) = \begin{cases} 1, & \text{se } \exists(\bar{q}_{cc}) \mid (\bar{q}_{cc} \varepsilon \bar{q}_{fnd}) \wedge (\forall k_i, g_i(d_j) = g_i(\bar{q}_{cc})) \\ 0, & \text{no caso contrário} \end{cases}$$

Ou também:

$$sim(q, d_j) = \begin{cases} 1 & \text{se } \exists(\bar{q}_{cc}) \mid (\bar{q}_{cc} \varepsilon \bar{q}_{fnd}) \wedge (\forall k_i \in \bar{q}_{cc} \rightarrow w_{ij} = 1 \wedge \\ & \forall \neg k_i \in \bar{q}_{cc} \rightarrow w_{ij} = 0) \\ 0 & \text{no outro caso} \end{cases}$$

$$sim(q, d_j) = \begin{cases} 1 & \text{se } \exists(\bar{q}_{cc}) \mid (\bar{q}_{cc} \varepsilon \bar{q}_{fnd}) \wedge \bar{q}_{cc} \subseteq \mathbf{R}(d_j) \\ 0 & \text{no outro caso} \end{cases}$$

Se a similaridade entre d_j e q é igual a 1, isto significa que o documento é relevante para a consulta q , mas poderia também não ser.

Se a similaridade entre d_j e q é igual a 0, isto significa que o documento não é relevante. Em outras palavras, o modelo booleano não admite *matching* (correspondência) parcial com a consulta, ou o documento é relevante ou não é o que constitui uma importante limitação desse modelo (BAEZA-YATES; RIBEIRO-NETO, 1999, p.27).

Outras limitações do modelo booleano são:

- “Nenhuma ordenação de documentos é fornecida;
- A expressão da necessidade de informação por uma expressão booleana é muitas vezes difícil;
- As consultas booleanas formuladas pelos usuários são freqüentemente simplistas;

- Em consequência: o modelo booleano retorna com poucos ou muitos documentos em resposta às consultas” (KAESTNER, 2006).

A seguir vamos mostrar um exemplo das limitações do modelo booleano.

Seja o vetor de consulta:

$q = \text{madeira and móveis and caminhão}$

E os documentos de nosso *corpus* (conjunto de documentos):

d_1 = a polícia apreendeu a carga de **madeira**.

d_2 = a encomenda de **móveis** chegou em um **caminhão**.

d_3 = a carga de **madeira** chegou em um **caminhão**.

No arquivo invertido, depois da eliminação das *stopwords* (palavras inúteis) esses documentos serão indexados da seguinte maneira:

polícia	→	d_1
apreendeu	→	d_1
carga	→	d_1, d_3
madeira	→	d_1, d_3
encomenda	→	d_2
móveis	→	d_2
chegou	→	d_2, d_3
caminhão	→	d_2, d_3

O termo índice **madeira** é contido nos documentos d_1 e d_3 , o termo índice **móveis** é contido no documento d_2 e o termo índice **caminhão** é contido nos documentos d_2, d_3 .

Conferimos agora o resultado de nossa busca:

$q = \text{madeira and móveis and caminhão}$

$q = \{d_1, d_3\} \cap \{d_2\} \cap \{d_2, d_3\}$

$\{d_2\} \cap \{d_2, d_3\} = \{d_2\}$

$q = \{d_1, d_3\} \cap \{d_2\} = \emptyset$

Embora os documentos d_2 e d_3 contenham dois dos termos de nossa pesquisa, eles não foram encontrados, devido ao fato de que o modelo booleano não fornece *matching* parcial.

Depois de ter analisado algumas das características do modelo booleano, no próximo item iremos descrever o modelo de espaço vetorial.

4.2 Modelo de espaço vetorial (Vector-space model – VSM)

O modelo de espaço vetorial (*Vector-space model – VSM* ou simplesmente *modelo vetorial*) é um dos mais eficazes para a extração e a recuperação de informação e, de modo geral para, o *KDT (Knowledge Discovery in Text)*.

Esse modelo foi proposto pela primeira vez por Salton et al. em 1975. De acordo com Baeza-Yates e Ribeiro-Neto (BAEZA-YATES; RIBEIRO-NETO, 1999, 27), o modelo vetorial pretende superar os limites do modelo booleano, em particular no que diz respeito à impossibilidade de ter (no modelo booleano) correspondência parcial e à dificuldade de ordenar os documentos encontrados.

Neste item serão analisados os aspectos principais do modelo de espaço vetorial, cuja proposta teórica dará sustentação ao nosso trabalho de pesquisa.

De acordo com Ackerman (2003), esse modelo se baseia nos seguintes cálculos:

1. O peso de cada palavra índice [termo índice] dentro de um determinado documento (no contexto preciso daquele documento) deve ser calculado para todos os documentos (N documentos). Este cálculo indica quanto uma palavra é importante dentro de um documento específico.
2. O peso de cada palavra índice deve ser calculado em relação a todo o conjunto de documentos (*document set*). Este cálculo indica quanto a palavra é importante em relação a todo o conjunto de documentos.
3. Para cada consulta (*query*), o vetor de consulta (*query vector*), é comparado com cada um dos vetores dos documentos [os vetores que representam os documentos] (*document vectors*). O resultado pode ser classificado, indicando qual documento se aproxima mais do vetor de consulta, e ordena os outros em função da distância.²

² 1. The weight of each index word across the entire document set needs to be calculated. This answers the question of how important the word is in the entire collection. 2. The weight of every index word within a given document (in the context of that document only) needs to be calculated for all N documents. This answers the question of how important the word is within a single document. 3. For any query, the query vector is compared to every one of the document vectors. The results can be ranked. This answers the question of which document

Para entendermos melhor como realizar esses cálculos, vejamos agora, na Tabela 4.2, os conceitos básicos e símbolos utilizados para descrever o modelo de espaço vetorial (ACKERMAN, 2003):

Tabela 4.2 Conceitos básicos e símbolos utilizados para descrever o modelo de espaço vetorial. (ACKERMAN, 2003)	
N	Número total de documentos do <i>corpus</i> /base de dados textuais. Se o <i>corpus</i> contém 1.100 documentos indexados, $N=1100$.
k_i	Um dos termos índices
n_i	Número de documentos que contêm o termo k_i . Se o número de documentos (N) é de 1100, então, n_i pode variar de 0 até 1100.
$freq_{i,j}$	Número de vezes em que uma dada palavra/termo (k_i) aparece em um dado documento d_j .
$w_{i,j}$	Peso do termo k_i no documento j (d_j).
$w_{i,q}$	Peso do termo k_i no vetor de consulta q (<i>query vector</i>).
$q = (w_{1,q}, w_{2,q}, \dots, w_{t,q})$	Vetor de consulta (<i>query vector</i>). Este vetor é constituído por coordenadas em um espaço t -dimensional, onde o número t de dimensões é dado pelo número de termos índices no conjunto de documentos N ; o vetor de consulta tem um peso para cada uma dessas palavras índices (termos índices).
$d_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$	Vetor do documento. Vetor que representa um documento. Se o <i>corpus</i> contém N documentos, existem N vetores representando os N documentos. Cada vetor tem um peso para cada uma das palavras/termos índices (<i>keyword</i>) no sistema de indexação. Se o conjunto de documentos é grande, muitos desses valores serão iguais a zero, devido ao fato de que muitos termos índices não aparecem em todos os documentos. Em termos matemáticos, esse fenômeno é chamado de esparsidade (<i>sparsity</i>).
$f_{i,j}$	Frequência normalizada de k_i em d_j
d_j	Um documento representativo, o j -ésimo

comes closest to the query, and ranks the others as to the closeness of the fit. (A ordem dos parágrafos no texto traduzido é 2,1,3).

4.2.1 Similaridade entre vetores

A similaridade entre dois vetores A e B é igual ao produto interno dos dois vetores dividido pelo produto dos módulos dos dois vetores:

$$\text{sim}(A, B) = \frac{A \bullet B}{|A| \times |B|} \quad (4.1)$$

No caso do espaço vetorial, são normalmente comparados o vetor de consulta (q) e um vetor de documento (d_j):

$$\text{sim}(d_j, q) = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \times |\vec{q}|} \quad (4.2)$$

Para encontrar qual documento se aproxima mais do vetor de consulta é necessário expandir essa fórmula introduzindo os pesos.

Seja:

O vetor do documento $d_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$

e o vetor de consulta $q = (w_{1,q}, w_{2,q}, \dots, w_{t,q})$

O produto interno³ de dois vetores A e B é igual a: $A \bullet B = a_1b_1 + a_2b_2 + a_3b_3 + \dots + a_nb_n$. (a_n , coordenadas do primeiro vetor, e b_n coordenadas do segundo vetor); assim o produto interno dos vetores d_j e q é igual a:

$$d_j \bullet q = (w_{1,j} * w_{1,q}) + (w_{2,j} * w_{2,q}) + \dots + (w_{n,j} * w_{n,q}) \quad (4.3)$$

Em seguida, substituindo a equação (4.3) na (4.2) obtemos a seguinte:

³ É bom lembrar que o resultado do produto interno de vetores é uma grandeza escalar.

$$\text{sim}(d, q) = \frac{(w_{1,j} * w_{1,q}) + (w_{2,j} * w_{2,q}) + \dots + (w_{n,j} * w_{n,q})}{|d| * |q|}$$

(4.4)

O módulo do vetor d é igual:

$$|d| = \sqrt{d \bullet d}$$

(4.5)

e o módulo do vetor q é igual a:

$$|q| = \sqrt{q \bullet q}$$

(4.6)

Substituindo a (4.5) e a (4.6) na (4.4), obtém-se:

$$\text{sim}(d, q) = \frac{(w_{1,j} * w_{1,q}) + (w_{2,j} * w_{2,q}) + \dots + (w_{n,j} * w_{n,q})}{\sqrt{d \bullet d} * \sqrt{q \bullet q}}$$

(4.7)

Para calcular o produto interno de um vetor por si mesmo utilizamos a equação (4.3):

$$d_j \bullet d_j = (w_{1,j} * w_{1,j}) + (w_{2,j} * w_{2,j}) + \dots + (w_{n,j} * w_{n,j})$$

(4.8)

ou

$$d_j \bullet d_j = w_{1,j}^2 + w_{2,j}^2 + \dots + w_{n,j}^2$$

(4.9)

ou mais simplesmente

$$\sum_{i=1}^t w_{i,j}^2$$

(4.10)

Podemos fazer o mesmo para o vetor \mathbf{q} e obter:

$$\mathbf{q} \bullet \mathbf{q} = (w_{1,q} * w_{1,q}) + (w_{2,q} * w_{2,q}) + \dots + (w_{n,q} * w_{n,q})$$

(4.11)

ou

$$\mathbf{q} \bullet \mathbf{q} = w_{1,q}^2 + w_{2,q}^2 + \dots + w_{n,q}^2$$

(4.12)

ou mais simplesmente

$$\sum_{i=1}^t w_{i,q}^2$$

(4.13)

Substituindo a equação (4.10) e a (4.13) na (4.7), obtém-se:

$$\text{sim}(d,q) = \frac{(w_{1,j} * w_{1,q}) + (w_{2,j} * w_{2,q}) + \dots + (w_{n,j} * w_{n,q})}{\sqrt{\sum_{i=1}^t w_{i,j}^2} * \sqrt{\sum_{i=1}^t w_{i,q}^2}}$$

(4.14)

O numerador da (4.14) também pode ser substituído pelo símbolo da série:

$$(w_{1,j} * w_{1,q}) + (w_{2,j} * w_{2,q}) + \dots + (w_{n,j} * w_{n,q}) =$$

$$\sum_{i=1}^t (w_{i,j} * w_{i,q}) \quad (4.15)$$

Assim, substituindo a equação (4.15) na (4.14), obtém-se a equação final para determinar a similaridade de um vetor de consulta com um vetor de documento que é:

$$sim(d,q) = \frac{\sum_{i=1}^t (w_{i,j} * w_{i,q})}{\sqrt{\sum_{i=1}^t w_{i,j}^2} * \sqrt{\sum_{i=1}^t w_{i,q}^2}} \quad (4.16)$$

Depois de ter visto como calcular a similaridade de dois vetores, passemos agora ao cálculo dos valores das palavras e dos pesos.

4.2.2 Importância das palavras e cálculo dos pesos: considerações gerais

O primeiro grupo de pesos que deve ser definido é o dos pesos usados no vetor de consulta. Este contém cada termo índice utilizado em todo o conjunto de documentos. O peso no vetor de consulta de um termo índice reflete sua importância no contexto de todo o conjunto de documentos. Quando o número N de documentos for grande, esse conjunto de pesos será razoavelmente estável: um novo documento pode ser acrescentado ao conjunto de documentos (do *corpus*) sem alterar os valores do vetor de consulta (ACKERMAN, 2003).

O número do segundo conjunto de pesos é potencialmente ilimitado. Para cada um dos documentos do *corpus* devemos calcular o peso de cada termo índice contido em cada documento. E cada documento tem um vetor contendo o peso para cada um dos termos índices que aparecem no documento (os termos que não aparecem em um documento tem peso 0). Ex: 1000 documentos, 1000 termos índices, devem ser calculados 1.000.000 de

pesos. E cada vez que um documento é acrescentado ao *corpus* (*document set*), deve ser calculado o conjunto de pesos das palavras contidas no novo documento (ACKERMAN, 2003).

Se o peso é relacionado à importância da palavra, devemos ter pesos que reflitam a importância da palavra em um único documento e sua importância no conjunto de documentos: uma palavra, que, por exemplo, aparece em todos os textos do conjunto, deverá ter uma importância relativamente baixa (ACKERMAN, 2003). Vejamos na Tabela 4.3 alguns critérios que deverão ser levados em consideração:

Tabela 4.3 Critérios para atribuição de pesos. (ACKERMAN, 2003)		
Importância em um único documento	Importância no conjunto de documentos	Peso
Alta	Alta	Muito alto
Baixa	Alta	Médio
Alta	Baixa	Médio
Baixa	Baixa	Muito baixo

Para chegar a determinar os pesos, levando em consideração a Tabela 4.3, utilizaremos dois tipos diferentes de frequências, que serão analisados nos próximos dois itens deste trabalho.

4.2.3 Frequência normalizada de um termo (*term frequency – tf*) k_i em d_j

A frequência absoluta de uma palavra é igual ao número de vezes que ela aparece em um documento (número de ocorrências). Esse tipo de frequência não é, no entanto, um bom indicador. Em textos longos teremos palavras com frequências absolutas elevadas, em textos de menor comprimento teremos frequências absolutas mais baixas. Para podermos tornar

esses valores mais homogêneos, utilizamos a frequência normalizada f_{ij} de uma palavra k_i (frequência normalizada de k_i em d_j), que é obtida dividindo a frequência absoluta de uma palavra em documento $freq_{i,j}$ pela frequência absoluta da palavra mais freqüente no documento $max_l freq_{l,j}$:

$$f_{i,j} = \frac{freq_{i,j}}{max_l freq_{l,j}}$$

(4.17)

Diferentemente da frequência absoluta, a frequência normalizada pode variar de 0 a 1 (Tabela 4.4):

- 0 indica ausência de uma palavra em um documento
- valores perto do zero indicam palavra de baixa frequência
- valores perto de um indicam palavras de uso comum ou muito freqüente no documento.

Tabela 4.4. Variação da frequência normalizada. (ACKERMAN, 2003)			
Palavra	$freq_{i,j}$	$max(freq_{i,j})$	$f_{i,j}$
interceptação	1	100	.01
decisão	10	100	0.1
de	50	100	0.5
o	100	100	1

4.2.4 Freqüência inversa de uma palavra em um documento (*Inverse document frequency*)

Para medir a freqüência de uma palavra no conjunto dos documentos do *corpus* é utilizada a **freqüência inversa** (*idf – inverse document frequency*), dada pela equação:

$$idf = \log (N/n_i). \quad (4.18)$$

Lembrando que:

N é o conjunto de todos os documentos (*corpus*)

n_i é o conjunto de documentos em que o termo k_i aparece.

Diferentemente da freqüência normalizada de uma palavra em um único texto, que cresce em função da freqüência da palavra naquele texto, a freqüência inversa de uma palavra diminui quando a palavra está presente em muitos textos (seu valor é zero quando está presente em todos).

Seja, N=1000; nesse caso o valor da freqüência inversa em nosso conjunto de documentos poderia variar conforme exemplificado na Tabela 4.5:

Tabela 4.5 Variação da freqüência inversa. (ACKERMAN, 2003)		
N	n_i	$\log(N/n_i)$
1000	1	3
1000	100	2
1000	500	.301029996
1000	1000	0

A frequência inversa teria valor 3 para palavras que aparecem em poucos documentos (ex. $n_i = 1$) e o valor tenderia a zero para palavras muito frequentes.

4.2.5 O cálculo dos pesos

A fórmula utilizada para o cálculo dos pesos nos documentos leva em conta tanto a frequência normalizada (Equação (4.17)) quanto a frequência inversa de uma palavra (Equação (4.18)), e é a seguinte:

$$w_{ij} = f_{ij} * \log \frac{N}{n_i} \quad (4.19)$$

O peso de um termo k_i em um documento d_j é igual à sua frequência normalizada (nf) multiplicada por sua frequência inversa no documento (*idf*). A Tabela 4.6 ilustra a variação dos pesos em função das duas frequências:

Tabela 4.6 Variação do peso em função da frequência normalizada e da frequência inversa. (ACKERMAN, 2003)		
Frequência normalizada	Frequência inversa	Peso
Alta	Alta	Muito alto
Baixa	Alta	Médio
Alta	Baixa	Médio
Baixa	Baixa	Muito baixo

Há várias maneiras de calcular os pesos. Baeza-Yates e Ribeiro-Neto (1999, p.30) indicam uma fórmula descrita por Salton e Buckley (1988) para calcular os pesos do vetor de consulta:

$$w_{i,q} = \left(0,5 + \frac{0,5 * freq_{i,q}}{\max_{l,q} freq_{l,q}} \right) * \log \frac{N}{n_i}$$

(4.20)

O peso de um termo k_i no vetor de consulta deve ser calculado utilizando a frequência normalizada do termo k_i no texto do documento de consulta q , corrigida pelo fator 0.5 (BAEZA-YATES; RIBEIRO-NETO, 1999, p.30). Isso é determinado pelo fato de que o valor da frequência normalizada é comumente baixo (ACKERMAN, 2003). No entanto, de acordo com Ackerman (2003), para o cálculo dos pesos no vetor de consulta deveria ser utilizada sempre a mesma fórmula empregada nos outros documentos (ver Equação (4.19)).

Neste capítulo estudamos dois dos modelos de recuperação de informações mais utilizados. Esses modelos nos mostraram como transformar textos/documentos em vetores, e como compará-los entre si. No próximo capítulo veremos que tipo de modelo semântico utilizar para nossa interface.

5 O modelo semântico de Pottier: os conceitos de sema, conceito, e noema

Este capítulo tem como objetivo estudar o modelo semântico de Bernard Pottier, que utilizaremos para implementar uma rede de Kohonen. A escolha do modelo desse autor deve-se ao fato de que tem características tais que permitem sua aplicação ao reconhecimento de padrões por meio de redes neurais. Nas páginas que se seguem, abordaremos vários aspectos lingüísticos ligados à obra de Pottier para melhor compreender seu modelo.

As definições de *percurso semasiológico* (ou do interpretante) e de *percurso onomasiológico* (ou do enunciador) serão o ponto de partida do estudo que faremos neste trabalho sobre o modelo de Pottier. No nível do “conceptual”, Pottier concebe, num primeiro momento, dois tipos de conceitos – os *conceitos gerais* ou *conceitos* e os *conceitos universais* ou *noemas*. Em *Sémantique générale* (POTTIER, 1992), o autor modifica o sentido que tinha atribuído a noema e a conceito em sua obra *Théorie et analyse en linguistique* (POTTIER, 1987). Depois de termos analisado o significado desses termos, veremos o modelo de signo de Pottier, examinando os conceitos de *significante*, *significado*, *sema*, *semema* (*semantema*, *classema*, *virtuema*). Para ilustrar o modelo de análise semântica proposto pelo referido autor, retomaremos, aqui, o exemplo que se tornou clássico, de *cadeira*, *poltrona*, *banquinho*, *sofá*, *pufe* e *assento* (POTTIER, 1964, p.125). Uma atenção particular será dada à maneira de descrever os semas utilizada por esse autor.

5.1 Os percursos onomasiológico e semasiológico

Bernard Pottier (em *Sémantique générale*) (1992) propõe-nos um interessante modelo do percurso gerativo da enunciação. Conforme nos posicionamos do ponto de vista do *enunciador* (emissor) ou do *interpretante* (receptor), teremos, respectivamente, o *percurso onomasiológico*, que vai da intenção de dizer ao enunciado, ou o *percurso semasiológico*, que vai do enunciado à sua interpretação. Pottier apresenta esquematicamente o percurso onomasiológico como mostrado na Figura 5.1 (POTTIER, 1992, p.16):

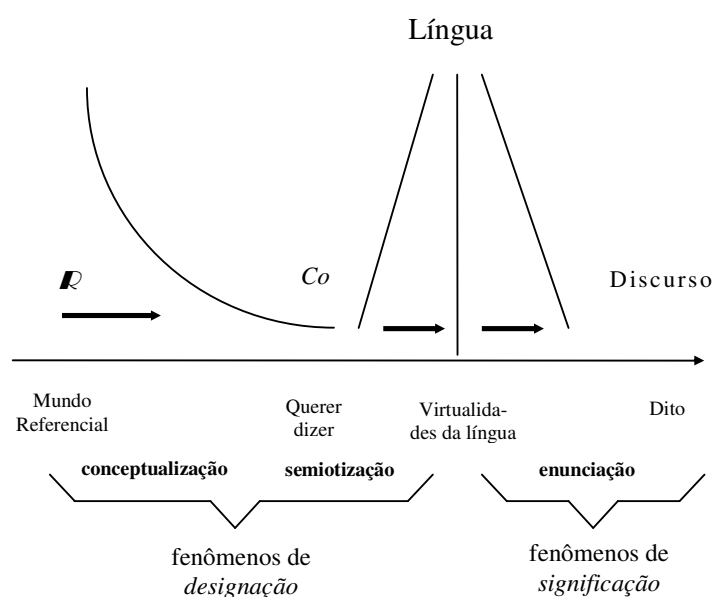


Figura 5.1.– Percurso onomasiológico. (POTTIER, 1992, p. 16)

O enunciador tem como ponto de partida o mundo referencial (R). Quando tem a intenção de dizer (o *querer dizer*) ele começa a conceptualizar sua intenção (Co). Essa conceptualização deve, então, ser colocada em signos através de um processo de semiotização. Este se faz através de meios fornecidos por um sistema semiótico. Em nosso caso, o sistema semiótico utilizado pelo enunciador é uma língua natural (LN). Para Pottier, as *virtualidades da língua* correspondem ao sistema semiótico da língua e aos “*mecanismos de enunciação que permitem as realizações discursivas*” (POTTIER, 1992, p.16)⁴.

⁴ « *mécanismes d'énonciation qui permettent les réalisations discursives* ».

A passagem da conceptualização à semiotização se faz através do que Pottier chama de *fenômenos de designação*, isto é, fenômenos pelos quais se estabelecem as relações entre o mundo referencial e os sistemas das línguas naturais. A *enunciação* é, assim, a passagem das virtualidades da língua aos discursos realizados portadores de significação, e se faz através do que o autor designa como os *fenômenos de significação*, ou melhor, os fenômenos pelos quais “os significados da língua se tornam significações em discurso” (POTTIER, 1992, p.20)⁵. Conseqüentemente, é muito importante, para Pottier, fazer também uma distinção entre os modelos frásticos em língua, que contêm significados em potencial, e os discursos realizados, que, ao contrário, são portadores de significação.

O *percurso semasiológico* é o do interpretante e acontece no sentido contrário ao do enunciador. Parte do discurso realizado (texto oral ou escrito) para chegar à compreensão. Na Figura 5.2 podemos ver o esquema correspondente ao percurso semasiológico (POTTIER, 1992, p.17):

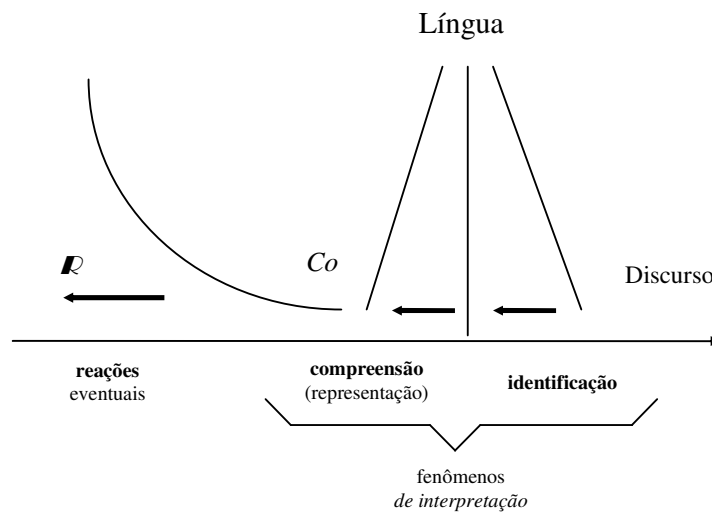


Figura 5.2. – Percurso semasiológico. (POTTIER, 1992, p.17)

O interpretante consegue compreender um texto (discurso) através da identificação e interpretação dos diferentes elementos discursivos que o compõem. Compreender um discurso é, para o interpretante, poder “*representá-lo mentalmente*”, “*conceptualizá-lo destacando-se*

⁵ « *les signifiés de langue deviennent des significations en discours* ».

rapidamente dos signos da língua natural que ele identificou e que lhe serviram de trampolim para a compreensão” (POTTIER, 1992, p.17)⁶.

Segundo Pottier, o mundo referencial (R), mesmo sendo o verdadeiro ponto de partida e de chegada dos dois percursos, não deve, obrigatoriamente, ser objeto de atenção do lingüista, “já que o conceptual (Co) pode por si só ser o ponto de partida para o enunciador e o ponto de chegada para o interpretante” (POTTIER, 1992, p.18)⁷. O conceptual é, para este autor, o “*lugar da representação mental que se tornou independente das línguas naturais e dos outros sistemas semiológicos, e sede da encenação*” (POTTIER, 1992, p.18)⁸.

Pottier considera a língua como um *saber* através do qual se realiza a transformação em signos e em esquemas, no caso do percurso do enunciador (onomasiológico), e a *interpretação*, no caso do percurso do interpretante (semasiológico).

O autor nos propõe um outro esquema para representar o percurso onomasiológico que traduz graficamente suas considerações (Pot92,18), conforme apresentado na Figura 5.3:

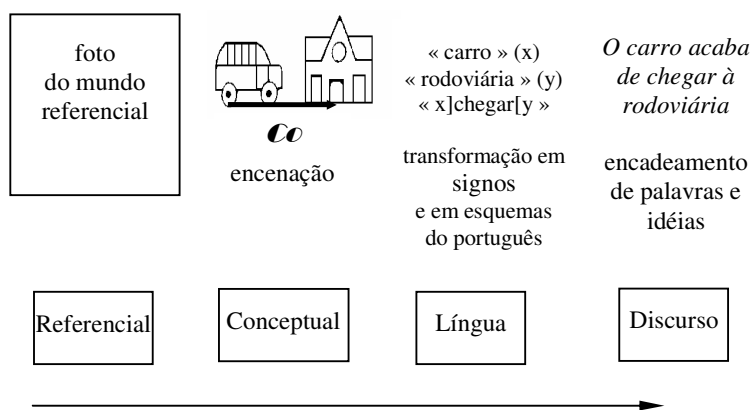


Figura 5.3. – Percurso onomasiológico. (POTTIER, 1992, p.18)

Nesse esquema, vemos a passagem da conceptualização ao discurso. As fases que nos interessam mais, para a continuação do nosso trabalho, são a fase da *encenação*, no nível conceptual, e a fase da *transformação em signos e em esquemas (processo de lexemização)*, no nível da língua. Depois de ter analisado os percursos onomasiológico e semasiológico, partindo do nível conceptual, veremos como se desenvolve o modelo de signo de Pottier.

⁶ «le représenter mentalement», «le conceptualiser en se détachant rapidement des signes de la langue naturelle qu’il a identifiés et qui lui ont servi de tremplin pour la compréhension».

⁷ « puisque le conceptuel (Co) peut à lui seul être le départ pour l’énonciateur et l’arrivée pour l’interprétant ».

⁸ « lieu de la représentation mentale devenu indépendant des langues naturelles et des autres systèmes sémiologiques, et siège de la mise en scène ».

5.2 Sema, conceito e noema no modelo de signo de Pottier

No nível conceptual, Pottier cria dois tipos de *conceitos* que permitem realizar a encenação: tratam-se dos *conceitos gerais* – ou *conceitos* – e *conceitos universais* – ou *noemas* (POTTIER, 1992, p.71).

Os conceitos gerais poderiam ser agrupados, segundo o autor, em quatro grandes “categorias” (POTTIER, 1992), como resumido na Tabela 5.1.:

Tabela 5.1. – Categorias de conceitos gerais. (POTTIER, 1992, p.72)

<i>Os seres</i>	<i>As coisas</i>	<i>As propriedades</i>	<i>As atividades</i>
homem	casa	leve	pescar
fêmea	sol	pesado	caminhar
peixe, etc.	água, etc.	jovem, etc.	olhar, etc.

Em todos esses conceitos haveria, segundo o autor, “*um componente natural (experiência comum) e um componente cultural (elementos específicos de acordo com o lugar e o tempo)*” (POTTIER, 1992, p.72)⁹.

Os *conceitos universais* ou *noemas* são, para Pottier, “*representações relacionais, abstraídas da experiência, mas cujos traços lingüísticos assumem formas muito variadas nas LN*” e “*um noema aparece, portanto, como uma relação abstrata universal sustentando as operações semânticas gerais das línguas, e é visualizável a fim de se aproximar o máximo possível da intuição de uma representação mental compartilhada*” (POTTIER, 1992, p.78).¹⁰

Para tentar entender o que Pottier compreende por *noema*, com base na Figura 5.4, mostraremos alguns exemplos de valores que o noema de “interioridade” pode assumir, segundo o autor:

⁹ « *une composante naturelle (expérience commune) et une composante culturelle (des éléments spécifiques selon le lieu et le temps)* ».

¹⁰ « *représentations relationnelles, abstraites de l'expérience, mais dont les traces linguistiques prennent des formes très variées dans les LN* » et « *un noème apparaît donc comme une relation abstraite universelle sous-tendant les opérations sémantiques générales des langues, et il est visualisable afin de se rapprocher le plus possible de l'intuition d'une représentation mentale partagée* ».

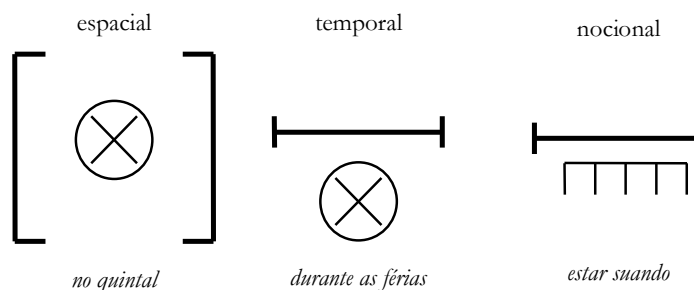


Figura 5.4. – Representações gráficas de alguns noemas. (POTTIER, 1992, p.78)

Pottier, em *Sémantique générale* (1992), modificou um pouco o sentido que tinha atribuído anteriormente a noema e a conceito. Tentaremos compreender essas “modificações”, fazendo uma retrospectiva de seu pensamento através da leitura de *Théorie et analyse en linguistique* (POTTIER, 1987). Nesse livro, Pottier define noema da seguinte maneira:

— o noema é um traço de sentido colocado independentemente de qualquer língua natural. Ele é absoluto (e não relativo a um conjunto) e sua existência é decidida pelo analista. Naturalmente, o lingüista criou os noemas baseado em certa experiência que ele teve das línguas naturais. Mas ele não os extrai automaticamente dos semas genéricos (ou de classe), mesmo que a afinidade seja evidente. (POTTIER, 1987, p.61)¹¹

O noema é, portanto, um traço de sentido no nível do conceptual, que é independente de qualquer língua natural, absoluto, e não relativo a um conjunto como será o sema (cf. *infra* p.80). Os noemas, segundo Pottier, são criados *ad hoc* pelo lingüista, pois não são observáveis diretamente. Pottier compara um noema a um quark: os prótons podem ser analisados em quarks, mesmo se ainda não conseguimos observá-los (POTTIER, 1987, p.67). Para o autor, os semas, por sua vez, “são somente elementos distintivos de um conjunto, são simplesmente elementos complexos com relação a possíveis elementos simples”, e, portanto, “um sema, unidade distintiva, é sempre analisável em noemas” (POTTIER, 1987, p.67)¹².

¹¹ « le noème est un trait de sens posé indépendamment de toute langue naturelle. Il est absolu (et non relatif a un ensemble) et son existence est décidée par l'analyste. Naturellement, le linguiste a créé les noèmes sur la base d'une certaine expérience qu'il a eue des langues naturelles. Mais il ne les tire pas automatiquement des sèmes génériques (ou de classe), bien que l'affinité soit évidente ».

¹² « ne sont que des éléments distinctifs d'un ensemble, ce sont tout naturellement des complexes par rapport à des simples possibles », ..., « un sème, unité distinctive, est toujours analysable en noèmes ».

Naquele momento, um conceito era, para Pottier, “*uma combinação de noemas*” (POTTIER, 1987, p.63). Na Figura 5.5 apresentamos graficamente os conceitos que podem se formar, “combinando” os noemas // limite // e // movimento// (POTTIER, 1987, p.63):

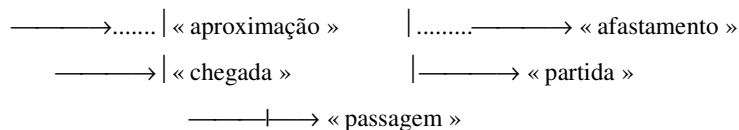


Figura 5.5. – Representação gráfica de alguns noemas. (POTTIER, 1992, p.17)

Em *Semantique générale* (POTTIER, 1992), a descrição do plano conceptual mudou, porém, com relação àquela da *Teoria e análise semântica* (POTTIER, 1987). O plano conceptual de *Teoria e análise*, que era caracterizado por noemas, considerados como traços de sentido absolutos, isto é, colocados independentemente de qualquer língua natural, torna-se, por sua vez, de certo modo, absoluto e independente de qualquer língua natural. Isso significa que a maneira de compreender e conceptualizar é a mesma para cada língua, cultura e indivíduo.

A nosso ver, tratava-se de posições parcialmente falsas. Karl Wilhelm von Humboldt (1767-1835) dizia que cada língua natural era caracterizada por uma *forma interna* (*innere Sprachform*) que exprime a concepção (ou a visão) do mundo própria da nação que a fala (*Weltanschauung*) (HUMBOLDT, 2000). Assim, sentimo-nos bastante próximos dessas posições e acreditamos que o plano conceptual de cada indivíduo é determinado, de certa maneira, por essa *Weltanschauung* própria a cada cultura. Certamente, as diferenças entre a *Weltanschauung* de um francês e a de um alemão não são obrigatoriamente enormes, mas são, com certeza, menos evidentes que as diferenças entre a de um francês e a de um bantu (África do Sul).

Em *Semantique générale* (POTTIER, 1992), no nível do plano conceptual, o componente cultural reaparece, portanto, nos conceitos gerais. Se esses conceitos têm um componente cultural, isso implica que não serão, obrigatoriamente, os mesmos, em todas as línguas e culturas. Por essa razão, também o plano conceptual não é mais absoluto. Tornou-se novamente relativo, e condicionado pelas dimensões espaço-temporais, assim como o plano da língua. Eis como Pottier voltou a questionar todo o plano conceptual de seu sistema teórico precedente. Se o plano conceptual não é absoluto, mas relativo, será possível realizar sistemas

de classificações de conceitos (ou sistemas nocionais), absolutos e independentes de qualquer língua e cultura? E ainda, esse “componente cultural”, que condiciona o plano conceptual, é o mesmo no interior de uma comunidade lingüística? Se fosse o caso, então seria possível realizar, no quadro de uma língua ou cultura, sistemas de classificação de conceitos com um valor universal para todos os indivíduos dessa língua ou cultura. No caso contrário, qualquer sistema de classificação de conceitos teria somente um valor relativo e jamais absoluto ou universal!

Após a colocação em esquemas, que se realiza em nível conceptual através de conceitos gerais e de noemas, veremos a transformação em signos (lexemização), que se realiza no nível da língua, por meio da escolha dos signos lingüísticos. Para o modelo do signo, preferimos utilizar o modelo proposto por Pottier, em *Lingüística geral: teoria e descrição* (POTTIER, 1985).

O autor toma, como ponto de partida, o modelo do signo de Saussure. Um *signo lingüístico* é, assim, composto por um *significado* e um *significante* (signo = significado + significante) e a relação entre significado e significante é a da *implicação dupla*, isto é, o significado e o significante são interdependentes (POTTIER, 1985, p.26):

significado <====> significante

O significado, por sua vez, é composto por uma “*substância (específica) e por uma forma (genérica), igualmente interdependentes*” (POTTIER, 1985, p.26)¹³, e assim temos os dois esquemas mostrados na Figura 5.6.

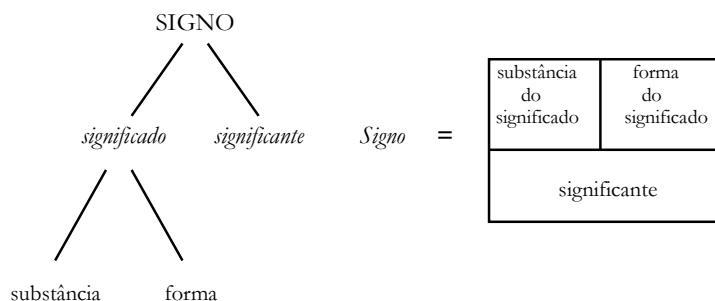


Figura 5.6. – O modelo de signo lingüístico de Pottier. (POTTIER, 1985, p.26)

¹³ « *substance (spécifique) et d'une forme (générique), également interdépendantes* ».

A substância do significado é “constituída por conjuntos de traços semânticos. A forma do significado, por traços classificatórios, que são a base de categorias (as ‘partes do discurso’ são um exemplo)” (POTTIER, 1985, p.26)¹⁴.

Isso implica que o plano da substância do significado é o da Semântica (**Se**), o plano da forma do significado é o da Sintaxe (**Si**) e o plano do significante é o da Significância (**Sa**) (POTTIER, 1985, p.27), como esquematizado na Figura 5.7.

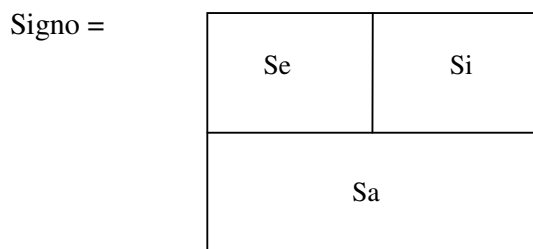


Figura 5.7. – Plano do significado e plano do significante. (POTTIER, 1985, p.77)

A substância do significado é chamada de *semema*. Cada traço semântico que compõe o semema é chamado de *sema*:

$$\text{Semema} = \{\text{sema}^1, \text{sema}^2, (\dots), \text{sema}^n\}$$

O sema é, assim, definido por Pottier, em *Teoria e Análise Lingüística* (POTTIER, 1987, p.61): “o sema é o traço distintivo semântico de um semema, relativamente a um pequeno conjunto de termos realmente disponíveis e possivelmente utilizáveis pelo locutor em uma dada circunstância de comunicação”¹⁵.

Segundo Pottier, os semas são, essencialmente, de dois tipos: *denotativos* e *conotativos*. Os semas denotativos são aqueles que determinam “de uma maneira estável e com ampla aceitação social a significação de um signo” (POTTIER, 1985, p.29)¹⁶. Os semas conotativos são aqueles que “caracterizam de uma maneira instável e freqüentemente individual a significação de um signo” (POTTIER, 1985, p.30)¹⁷.

¹⁴ « constituée par des ensembles de traits sémantiques. La forme du signifié par des traits classificatoires qui sont la base de catégories (les "parties du discours" en sont un exemple) ».

¹⁵ « le sème est le trait distinctif sémantique d'un sémème, relativement à un petit ensemble de termes réellement disponibles et vraisemblablement utilisables chez le locuteur dans une circonstance donnée de communication ».

¹⁶ « d'une façon stable et avec une vaste assise sociale la signification d'un signe ».

¹⁷ « caractérisent d'une façon instable et souvent individuelle la signification d'un signe ».

Os semas denotativos se dividem, por sua vez, em *semas específicos* e *semas genéricos*. Os semas específicos são aqueles que permitem distinguir entre eles sememas vizinhos, e os semas genéricos indicam “*uma categoria geral*” (POTTIER, 1985, p.30).

O conjunto de semas específicos constitui o *semantema*; o conjunto de semas genéricos, o *clasema*, e o conjunto de semas conotativos, o *virtuema*. Os dois esquemas mostrados na Figura 5.8 resumem todas essas considerações sobre a substância do significado (POTTIER, 1985, p.30):

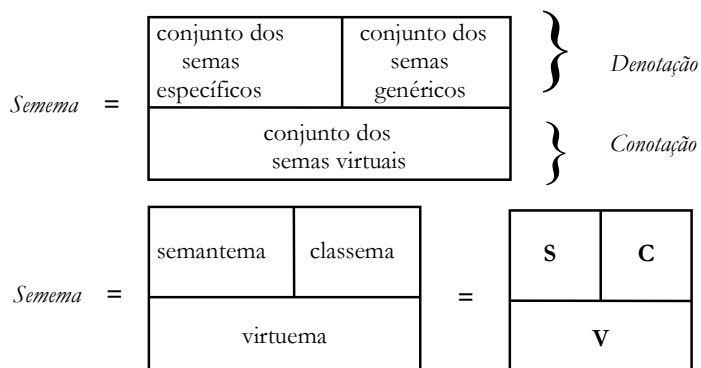


Figura 5.8. – O semema: semantema, clasema e virtuema. (POTTIER, 1985, p.30)

Para entender melhor o que Pottier compreende por sema, retomamos, aqui, o exemplo dado pelo autor, que se tornou clássico, de *cadeira*, *poltrona*, *banquinho*, *sofá*, *pufe* e *assento* (POTTIER, 1964, p.125). Para definir a substância do significado (semema) desses morfemas (palavras), como mostrado na Tabela 5.2, o autor utiliza os semas seguintes:

semas :

/« com encosto »/	= s ₁
/« com pé(s) »/	= s ₂
/« para 1 pessoa »/	= s ₃
/« para se sentar »/	= s ₄
/« com braços »/	= s ₅
/« com material rígido »/	= s ₆

Tabela 5.2. Análise semântica de um conjunto de palavras. (POTTIER, 1964, p.125)

semas										semema			morfema	
s ₁	+	s ₂	+	s ₃	+	s ₄	+			s ₆	=	S ₁	=	cadeira
s ₁	+	s ₂	+	s ₃	+	s ₄	+	s ₅	+	s ₆	=	S ₂	=	poltrona
		s ₂	+	s ₃	+	s ₄	+			s ₆	=	S ₃	=	banquinho
s ₁	+	s ₂	+			s ₄	+	s ₅	+	s ₆	=	S ₄	=	sofá
		s ₂	+	s ₃	+	s ₄					=	S ₅	=	pufe
		s ₂	+		+	s ₄					=	AS	=	assento
										arquissemema			arquimorfema	

O arquissemema é obtido, segundo Pottier, pela intersecção dos diferentes sememas:

$$S_1 \cap S_2 \cap S_3 \cap S_4 \cap S_5 = s_2 + s_4 = \text{arquissemema(AS)}.$$

O arquissemema (AS) é, assim, constituído pela soma de dois semas: (/“com pé(s)”/ + /“para se sentar”). Nesse caso específico, a essa substância de significado corresponde, na língua portuguesa, o morfema *assento*. Assim, *assento* é o *arquimorfema* desse conjunto de morfemas.

No que concerne aos *semas*, uma questão é fundamental para nosso trabalho: compreender como Pottier os “explicita”. Vejamos, em algumas palavras, o pensamento do autor:

Do nosso ponto de vista, o sema tem que ser expresso com tantas palavras da língua natural quantas forem necessárias para destacar bem o traço distintivo relativo ao conjunto considerado. A denominação do sema é um *discurso perifrástico com vocação metalingüística (ad hoc)*. (POTTIER, 1987, p.67)¹⁸

Os semas são, portanto, expressos por um “*discurso perifrástico com vocação metalingüística*”, criado *ad hoc* pelo lingüista. Trata-se de um discurso com “*vocação metalingüística*”, e não simplesmente metalingüístico, pois para Pottier o sema pode ser percebido no nível da língua e observado pelo lingüista, assim como os prótons e os nêutrons, no átomo (POTTIER, 1987, p.67). É interessante evidenciar o “*princípio da economia*” presente na maneira de denominar os semas, utilizado pelo autor (POTTIER, 1987, p.67).

¹⁸ « De notre point de vue, le sème doit se dire avec autant de mots de la langue naturelle qu'il faut pour bien mettre en relief le trait distinctif relatif à l'ensemble considéré. La dénomination du sème est un discours pé-riphrastique à vocation métalinguistique (*ad hoc*) ».

Entre as perífrases possíveis, será necessário utilizar aquelas que comportam o número exato de palavras (nem mais, nem menos).

Nesse capítulo estudamos o modelo semântico do lingüista francês Bernard Pottier. Esse modelo nos mostra como analisar, do ponto de vista semântico, o conceito (a substância do significado) de uma palavra, dividindo-a em traços semânticos que Pottier chama de semas. No próximo capítulo veremos como utilizar quanto proposto por Pottier para que nossa interface possa reconhecer padrões lexicais.

6 A implementação da rede

Ao longo dos capítulos precedentes adquirimos os conhecimentos necessários para implementarmos, neste capítulo, uma rede de Kohonen capaz de reconhecer padrões lexicais, que era o objetivo principal de nosso trabalho de pesquisa.

6.1 O programa Nenet

Uma Rede de Kohonen pode ser implementada de muitas maneiras diferentes e há vários simuladores distribuídos gratuitamente como *shareware* ou *freeware*, além de outros que são vendidos no mercado. Dentre esses, testamos alguns e acabamos escolhendo o programa finlandês Nenet 1.1 (NENET TEAM, 1999), desenvolvido pelo *Nenet Team* (da Universidade de Tecnologia de Helsinki), composto atualmente por cinco pesquisadores: Petri Hassinen, Jyri Elomaa, Jukka Rönkkö, Jokko Halme e Petri Hodju. O projeto começou em 1996, no quadro do curso ‘*Software Project*’ na Universidade de Tecnologia de Helsinki. A versão atualmente distribuída é a 1.1., que adotamos em nosso trabalho. A versão que utilizamos é *shareware* e pode ser baixada do *site*:

<http://koti.mbnet.fi/~phodju/nenet/Nenet/General.html>.

6.2 Da análise semântica aos padrões

Nosso ponto de partida, antes da implementação de uma rede de Kohonen, será o modelo de Pottier que nos vimos no Capítulo 5, p.80. Pottier analisa semanticamente cinco termos (*cadeira, poltrona, banquinho, sofá, pufe*), e identifica como arquisemema, o termo *assento*. Em uma estrutura a árvore conceptual, sistema nocional ou ontologia, esse termo seria o termo superordenado ou hiperônimo:

- 1. assento
 - 1.1. banquinho
 - 1.2. cadeira
 - 1.3. poltrona
 - 1.4. pufe
 - 1.5. sofá

O símbolo que precede o termo é chamado de símbolo de classificação. Os conceitos desses termos são descritos por um conjunto de seis semas (ou traços semânticos):

/« com encosto »/	= s ₁
/« com pé(s) »/	= s ₂
/« para 1 pessoa »/	= s ₃
/« para se sentar »/	= s ₄
/« com braços »/	= s ₅
/« com material rígido »/	= s ₆

Para que nossa rede de Kohonen possa classificar os termos (conceitos) analisados por Pottier, teremos de transformá-los em vetores. O conjunto de termos é composto pela combinação de seis semas. Poderemos, assim, transformar esses termos em vetores de seis dimensões, indicando a presença ou ausência de um dos seis semas. A Tabela 6.1 mostra essa transformação.

Tabela 6.1 Transformação em vetores de um conjunto de palavras						
	S₁	S₂	S₃	S₄	S₅	S₆
	/« com encosto »/	/« com pé(s) »/	/« para 1 pessoa »/	/« para se sentar »/	/« com braços »/	/« com material rígido »/
1. assento	0.0	1.0	0.0	1.0	0.0	0.0
1.1. banquinho	0.0	1.0	1.0	1.0	0.0	1.0
1.2. cadeira	1.0	1.0	1.0	1.0	0.0	1.0
1.3. poltrona	1.0	1.0	1.0	1.0	1.0	1.0
1.4. pufe	0.0	1.0	1.0	1.0	0.0	0.0
1.5. sofá	1.0	1.0	0.0	1.0	1.0	1.0

O número 1 indica a presença do sema, o zero indica ausência. No Capítulo 4, vimos como transformar textos em vetores, analisando dois modelos de recuperação de informações, o modelo booleano e modelo de espaço vetorial. Nossos vetores terão peso binário, com valores 1 e 0, como no modelo booleano.

Para que nossos vetores possam ser submetidos à rede, devem ser adaptados ao formato exigido pelo programa Nenet. Tratando-se de uma rede de Kohonen, utilizaremos os mesmos vetores para inicializar a rede, para treiná-la e, enfim, para testá-la (**Tabela 6.2**). No último conjunto de vetores, acrescentamos, entre parênteses, a etiqueta correspondente a cada um de nossos termos (símbolo de classificação e termo).

Tabela 6.2. Vetores de inicialização, treinamento e de teste									
Vetores de inicialização			Vetores de treinamento			Vetores para testar a rede			
0.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	“1. assento”
0.0	1.0	1.0	1.0	0.0	1.0	0.0	1.0	1.0	“1.1. banquinho”
1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	“1.2. cadeira”
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	“1.3. poltrona”
0.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	1.0	“1.4. pufe”
1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	“1.5. sofá”

Depois de transformar nossos termos em vetores, implementaremos agora uma pequena rede de Kohonen, com o programa Nenet, para ver como a rede se comporta no agrupamento desses termos. Do ponto de vista lingüístico, tendo em vista o fato de que

estamos submetendo à rede o semema de cada termo (o conjunto de semas de seu conceito), a rede vai, de fato, agrupar conceitos.

Fizemos vários testes e vimos que a rede ideal para classificar corretamente nossos vetores era uma rede de 6 x 6 neurônios. Do ponto de vista gráfico, pareceu-nos mais eficaz o hexágono para representar um neurônio.

Os parâmetros iniciais escolhidos para ativar a rede (como mostrado na Figura 6.1) foram os seguintes:

- a) Dimensões da rede: 6 x 6 neurônios
- c) Representação hexagonal de um neurônio (*Topology*)
- b) A função de vizinhança (*Neighbourhood Function*) escolhida é a de bolha (*Bubble*).
- c) O tipo de inicialização (*Initialization type*) é casual (*random*), e o valor “*random seed*” aconselhado pelos autores do *software* é de 123.
- d) Tendo em vista que nossos vetores já estão normalizados, não escolhemos nenhum tipo de pré-processamento dos vetores (*Preprocessing Method*).
- e) O arquivo *vetoresinittrein.dat*, contém os vetores de inicialização do sistema. Tal arquivo será utilizado também, sucessivamente, para treinar a rede (*Initialization & Training Data File*).

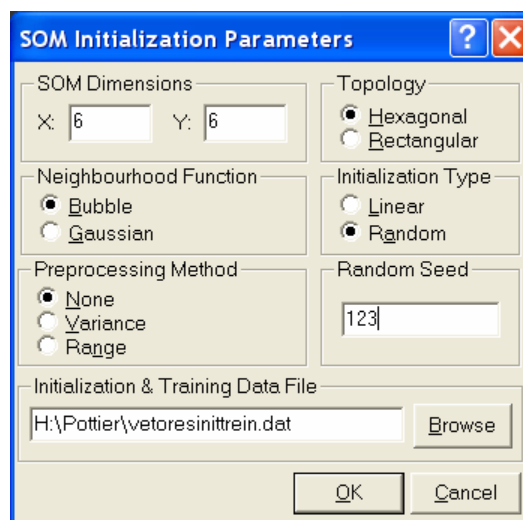


Figura 6.1 – Escolha dos parâmetros de inicialização da rede

Uma vez inicializada a rede se apresenta como ilustrado pela Figura 6.2. Como podemos ver foram criados e inicializados 36 neurônios.

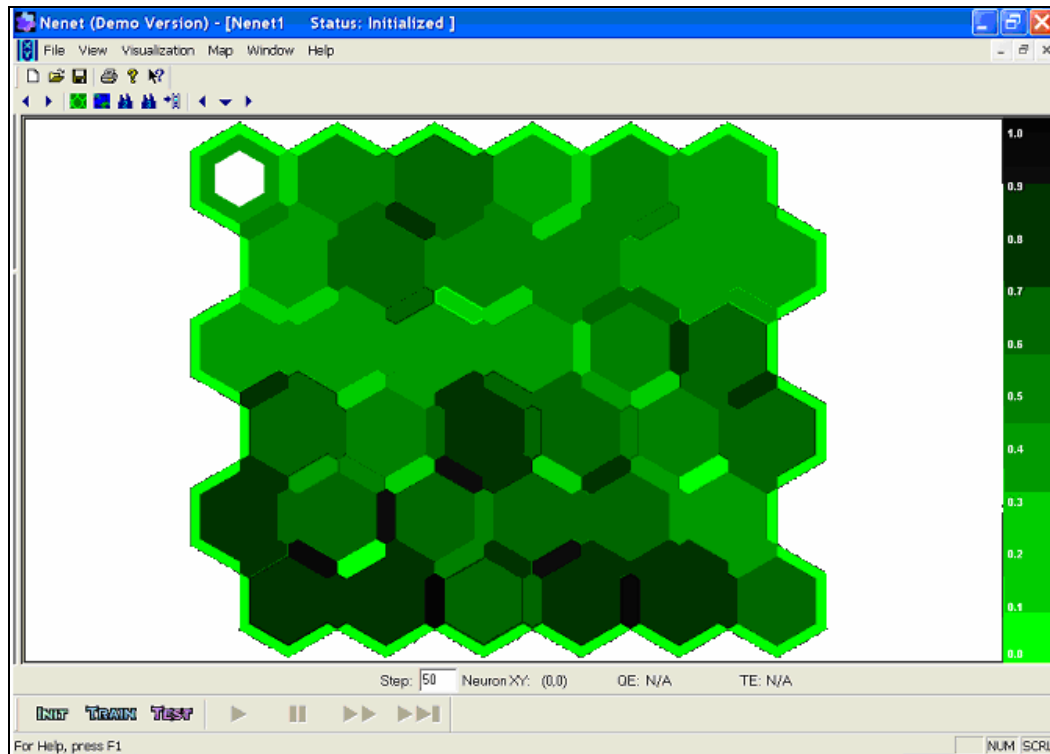


Figura 6.2 – Representação gráfica da rede inicializada

Depois da inicialização os vetores de pesos de todos os neurônios da rede são inicializados com valores aleatórios. Porém, o programa, não permite ver esses valores e executa a inicialização da rede juntamente aos vetores de inicialização. O conteúdo do arquivo *vetoresinittrein.dat* é o seguinte:

```
6
0.0  1.0  0.0  1.0  0.0  0.0
0.0  1.0  1.0  1.0  0.0  1.0
1.0  1.0  1.0  1.0  0.0  1.0
1.0  1.0  1.0  1.0  1.0  1.0
0.0  1.0  1.0  1.0  0.0  0.0
1.0  1.0  0.0  1.0  1.0  1.0
```

Na primeira linha se coloca o número de pesos de cada vetor. Em nosso caso, 6. Esse número será utilizado também para inicializar os neurônios da rede que terão “vetores pesos

sinápticos” compostos de seis pesos. O vetor pesos sinápticos do primeiro neurônio da rede, indicado por um pequeno hexágono branco, é apresentado na Figura 6.3:

Prm Level	Vector Values
1	0.086334
2	1.000000
3	0.672180
4	1.000000
5	0.637299
6	0.787750

Figura 6.3. – Valores dos pesos sinápticos do vetor do primeiro neurônio da rede.

Para visualizar os valores do vetor pesos sinápticos de cada neurônio é necessário utilizar o menu *View*, e escolher *Labels and vectors*.

Em outros simuladores de Redes Neurais de Kohonen, como o SNNS (*Stuttgart Neural Network Simulator*) (SNNS-GROUP, 1998), a rede é inicializada sem vetores de inicialização. Uma vez desenhada a rede, o programa inicializa os vetores pesos sinápticos. Além disso, o que tem de diferente, no programa Nenet, em relação a outros simuladores, inclusive o SNNS, é que a representação gráfica da rede de Kohonen apresenta somente uma camada de neurônios, sem mostrar nada mais, além disso. A Figura 6.4 mostra uma rede de Kohonen de 6 x 6 neurônios criada com o programa SNNS. Os primeiros seis quadradinhos da rede representam os componentes dos vetores de entrada (os pesos) e indicam também o número de componentes do vetor pesos sinápticos de cada neurônio.

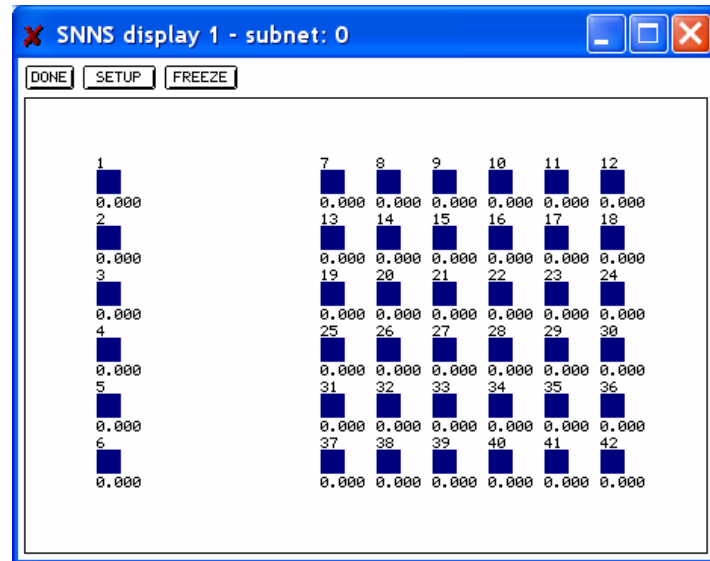


Figura 6.4. – Rede de 6 x 6 neurônios criada com o programa SNNS.

Depois de ter inicializado a rede, utilizamos, em seguida, o mesmo conjunto de vetores para treiná-la, isso se deve ao fato de termos poucos padrões a serem submetidos à rede. Os parâmetros utilizados para o treinamento, como podemos ver na Figura 6.5, são:

- a) Os parâmetros de aprendizado (*Learning Parameter*). Os autores do programa aconselham a utilizar uma taxa de aprendizagem de 0.05, e como função de adaptação dos pesos uma função de tipo linear (*Function Type*).
- b) O parâmetro *neighbourhood radius* indica a extensão (tamanho) da vizinhança no momento inicial do treinamento (*start = 10*) e no fim do treinamento (*stop = 1*).
- c) Devido ao número reduzido de padrões, escolhemos um treinamento de mil ciclos (*Training Length*).
- d) Os métodos de pré-processamento (*Preprocessing Method*) podem ser escolhidos somente na fase de inicialização. Tendo em vista que optamos por não escolher nenhum método de pré-processamento, a caixa de seleção fica inativa.
- e) O arquivo contendo os vetores de treinamento é o *vetoresinittrein.dat*, que utilizamos também para inicializar a rede (*Training Data File*).

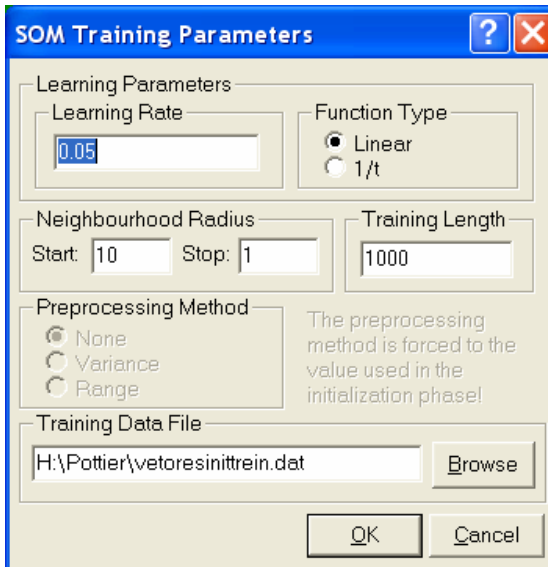




Figura 6.5. – Escolha dos parâmetros para treinar a rede

O treinamento pode ser efetuado de maneira contínua, apertando a tecla *play*  (*animate*), ou seqüencialmente, acionando a tecla *avante*  (*one step forward*), escolhendo os números de passos (*steps*) que o programa deverá efetuar cada vez que essa tecla for acionada.

Na fase de treinamento ocorrem os três processos característicos da rede de Kohonen, definidos por Haykin (2001, p.487): competitivo, cooperativo e adaptativo (cf. p.41). Por meio da distância euclidiana o programa Nenet compara os vetores de treinamento com os vetores de referências (vetores pesos sinápticos) dos neurônios da rede (processo competitivo). É interessante lembrar que no modelo de espaço vetorial a fórmula mais utilizada para comparar dois vetores é a mesma (cf. p. 42), similaridade entre vetores). Assim, a rede de Kohonen se comporta, em suas fases iniciais, como o modelo de espaço vetorial, que nós estudamos no item 4.2.

Em seguida o neurônio vencedor estabelece uma vizinhança topológica de neurônios (*neighbourhood radius*) que serão excitados (atraídos) pelo neurônio vencedor. A amplitude da atração dos neurônios é determinada pela taxa de aprendizagem (processo cooperativo).

Finalmente, os pesos dos vetores dos neurônios vizinhos são adaptados (processo adaptativo) e a vizinhança topológica diminui em função da taxa de aprendizagem (*learning*

rate). O parâmetro *learning rate* escolhido determina a velocidade com que a taxa de aprendizagem decresce (a zero).

A Figura 6.6. mostra a rede depois da fase de treinamento. Cores mais escuras mostram maior atividade de alguns neurônios da rede em relação aos demais.

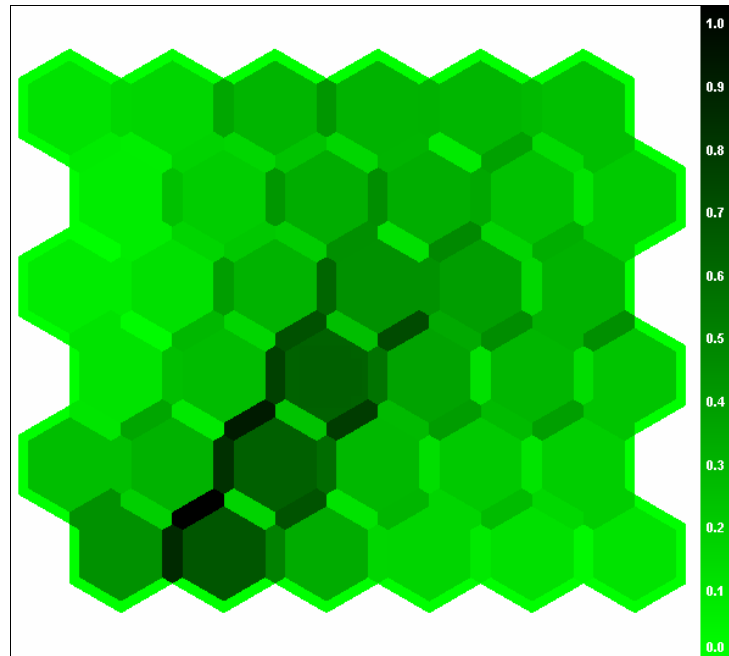


Figura 6.6. – Representação gráfica da rede depois do treinamento

Tendo em vista que não colocamos nenhuma etiqueta em nossos padrões de inicialização e de treinamento, não efetuaremos análise alguma sobre a distribuição dos padrões na rede.

Finalmente, passaremos à fase de teste em que os padrões serão apresentados à rede, um por um. Os parâmetros utilizados para testar a rede, como mostrado pela Figura 6.7, são:

- a) A caixa “*Clear BMU [Best Matching Unit]*¹⁹ Hits” reinicializa (a zero) todos os neurônios vencedores. Caso não seja ativada, na fase de teste os novos neurônios vencedores se acrescentarão aos neurônios vencedores existentes.

¹⁹ *Best match unit* é um neurônio vencedor, escolhido com o critério da distância euclidiana.

b) “*Use Automatic Labelling*” permite colocar sobre os neurônios, automaticamente, as etiquetas contidas nos vetores. No que diz respeito ao método de pré-processamento, esse continua sendo o mesmo escolhido na fase de inicialização e não pode ser mudado (cf. *supra* p. 87).

e) O arquivo contendo os vetores de teste é o *vetoresparateste.dat*, que utilizamos também para inicializar a rede (*Training Data File*):

```

6
0.0  1.0  0.0  1.0  0.0  0.0  "1. assento"
0.0  1.0  1.0  1.0  0.0  1.0  "1.1. banquinho"
1.0  1.0  1.0  1.0  0.0  1.0  "1.2. cadeira"
1.0  1.0  1.0  1.0  1.0  1.0  "1.3. poltrona"
0.0  1.0  1.0  1.0  0.0  0.0  "1.4. pufe"
1.0  1.0  0.0  1.0  1.0  1.0  "1.5. sofá"

```

Entre aspas é escrita a etiqueta do vetor, que será colocada, depois da fase de teste, sobre o neurônio vencedor.

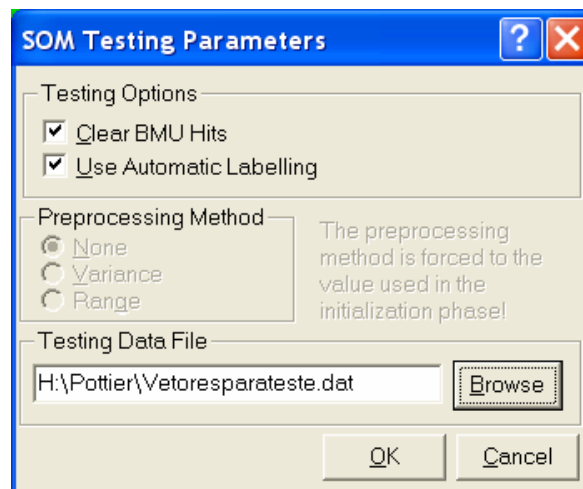


Figura 6.7. – Escolha dos parâmetros para testar a rede

Depois de ter apresentado os vetores de teste à rede, ela se apresenta como mostrado na Figura 6.8. Os termos se distribuem segundo uma lógica bem precisa. “Pufe” é o que se encontra mais perto de “assento”. O conceito desses dois termos difere somente por um traço semântico: /« para 1 pessoa »/. “Cadeira” e “banquinho” se encontram perto um do outro enquanto as duas noções são separadas somente por um traço semântico: “banquinho” não

tem /« com encosto »/. Ao mesmo tempo, esse par de termos é diferente de “pufe” e “assento”. “Poltrona” e “sofá” também estão perto um do outro, o único traço semântico que os separa é /« para 1 pessoa »/.

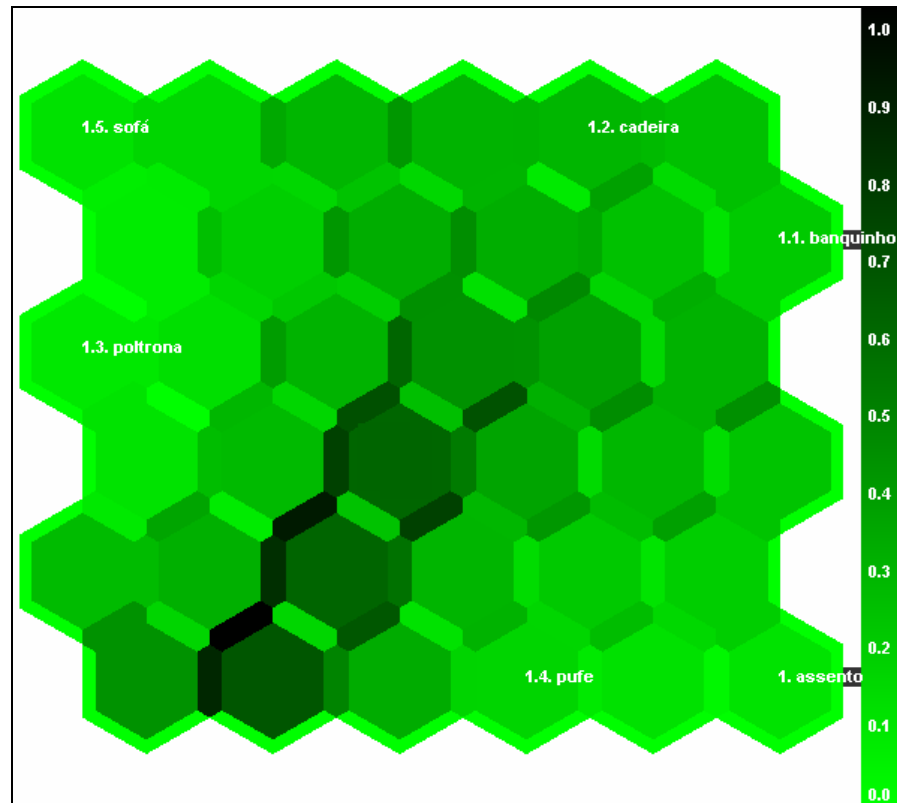


Figura 6.8. – Representação gráfica da rede depois da fase de teste

É interessante observar que “poltrona” e “sofá” são os conceitos que se situam mais longe de “assento”: “poltrona” contém todos os traços semânticos do conjunto de termos analisados, “sofá”, cinco, enquanto que o hiperônimo “assento” é o que contém menos traços, somente dois.

Na fase de teste, a rede de Kohonen utiliza o critério da distância euclidiana para achar os neurônios vencedores: nesse caso os vetores de teste são comparados com os demais vetores dos 36 neurônios da rede.

O programa Nenet permite também representar graficamente as relações entre os diferentes parâmetros dos neurônios em um histograma em 2 dimensões. Essas relações subtendem as relações semânticas entre os termos/conceitos distribuídos na rede.

Para mostrar as relações entres os diferentes parâmetros da rede é necessário seleccionar um único parâmetro no menu *Labels and vectors*. Na Figura 6.9. é representado graficamente o parâmetro 1, a primeira dimensão de nosso vetor, que correspondia ao sema /« com encosto »/): “assento”, “banquinho” e “pufe” tinham esse parâmetro inicial igual a zero.

Depois da modificação dos pesos dos vetores, as cores mostram as similaridades entre esses três termos. À direita do mapa encontra-se a paleta de cores utilizada para a representação. Tendo em vista que nossos vetores eram normalizados, temos valores de zero a um: cores mais escuras para valores mais baixos, cores mais vivas para valores mais altos.

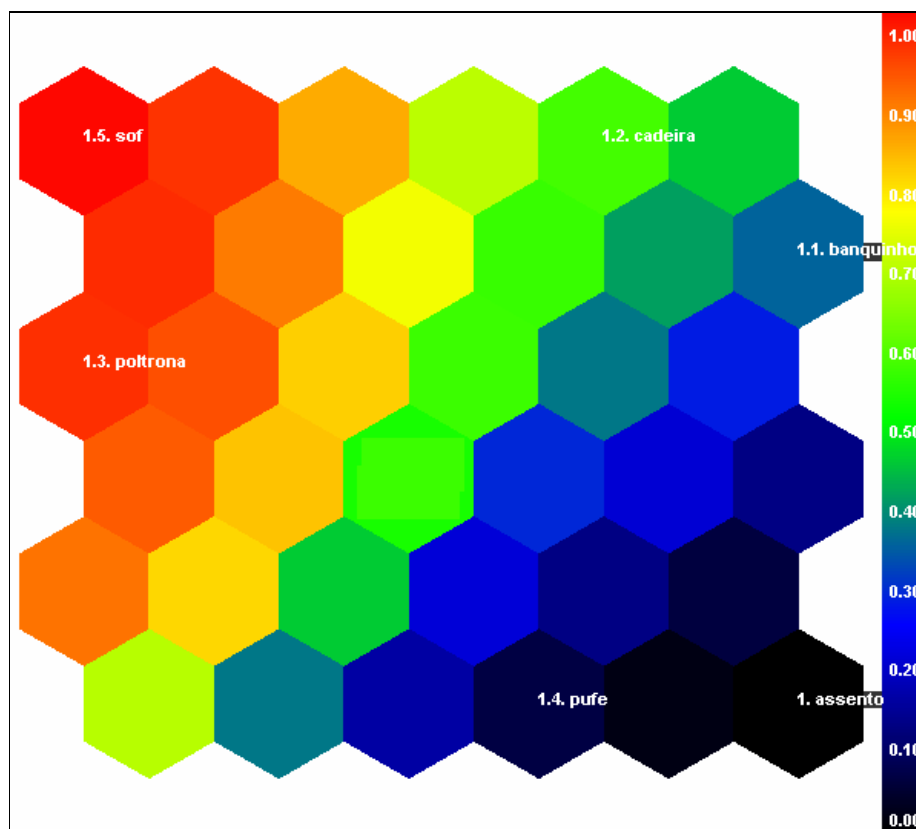


Figura 6.9. – Representação gráfica do primeiro parâmetro dos vetores de teste

É possível repetir esse tipo de representação utilizando os demais parâmetros dos vetores de teste. A Figura 6.10 mostra o que acontece se se seleciona o parâmetro 6 (/« com material rígido »/). Nesse caso, as cores mais vivas do verde até o vermelho indicam a presença do traço semântico inicial (o parâmetro 6), cores mais escuras, a ausência.

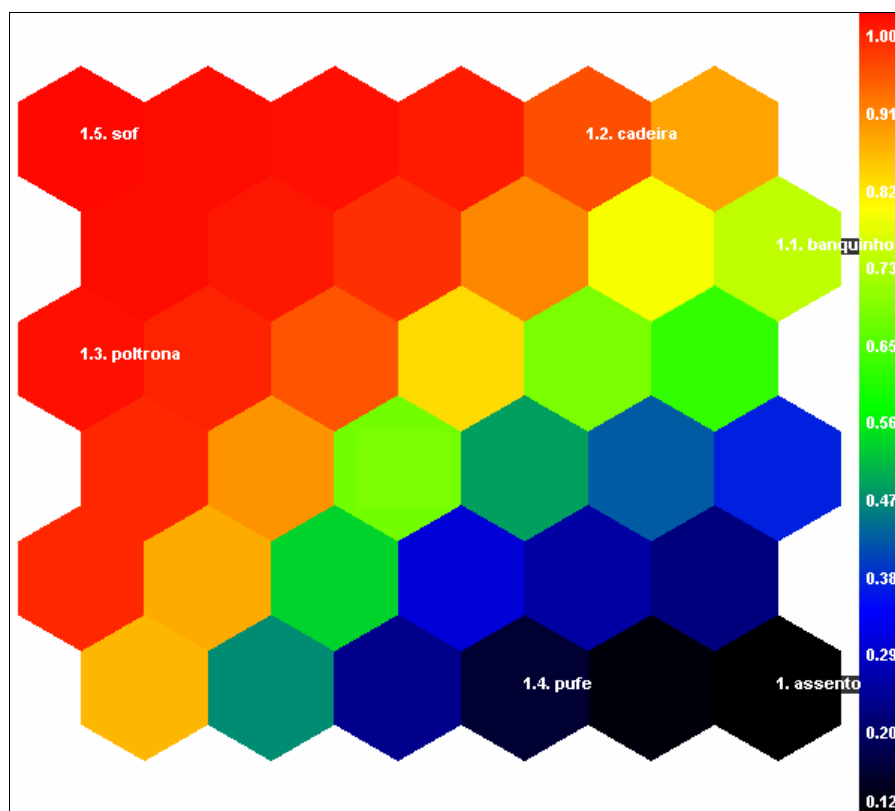


Figura 6.10. – Representação gráfica do sexto parâmetro dos vetores de teste

Depois de ter escolhido parâmetros que não são comuns a todos os termos de nosso conjunto de partida, vamos ver agora, na Figura 6.11, como se apresenta a rede escolhendo o parâmetro 2 (/« com pé(s) »/ ou o parâmetro 4 (/« para se sentar »/), comuns à todos os termos. A rede tem agora uma única cor, a cor preta, mostrando que o parâmetro 2 e o parâmetro 4 eram comuns a todos os termos.

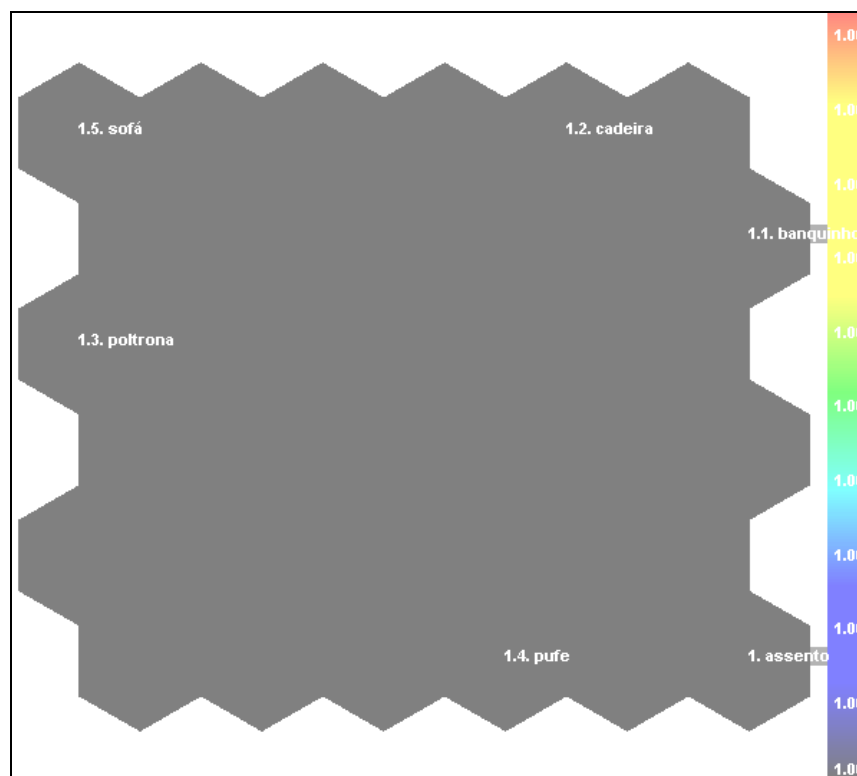


Figura 6.11. – Representação gráfica do segundo e do quarto parâmetro dos vetores de teste

De modo geral, podemos afirmar que a rede mostrou ter agrupado corretamente os padrões apresentados.

6.3 Testando a rede

A rede de Kohonen, como já dissemos, utiliza um tipo de treinamento competitivo e não-supervisionado, por meio do qual organiza os vetores (dados) de entrada em agrupamentos (*clusters*). Uma vez treinada a rede é capaz de classificar novos padrões, comparando-os com os padrões previamente utilizados. Para ver se a rede reconhece e classifica corretamente um padrão, apresentamos agora o vetor $\{1\ 1\ 1\ 1\ 0\ 1\}$ (Tabela 6.3):

Tabela 6.3. – Vetor de teste de padrão desconhecido						
	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆
	/« com encosto »/	/« com pé(s) »/	/« para 1 pessoa »/	/« para se sentar »/	/« com braços »/	/« com material rígido »/
“cadeira ?”	1.0	1.0	1.0	1.0	0.0	1.0

Submetemos à rede um arquivo contendo este vetor. Para verificar se a rede classifica corretamente o padrão submetido, não limpamos as etiquetas dos vetores de teste utilizados previamente. A Figura 6.12 mostra que o padrão submetido é reconhecido como cadeira: o neurônio vencedor é o mesmo que o da fase de teste precedente.

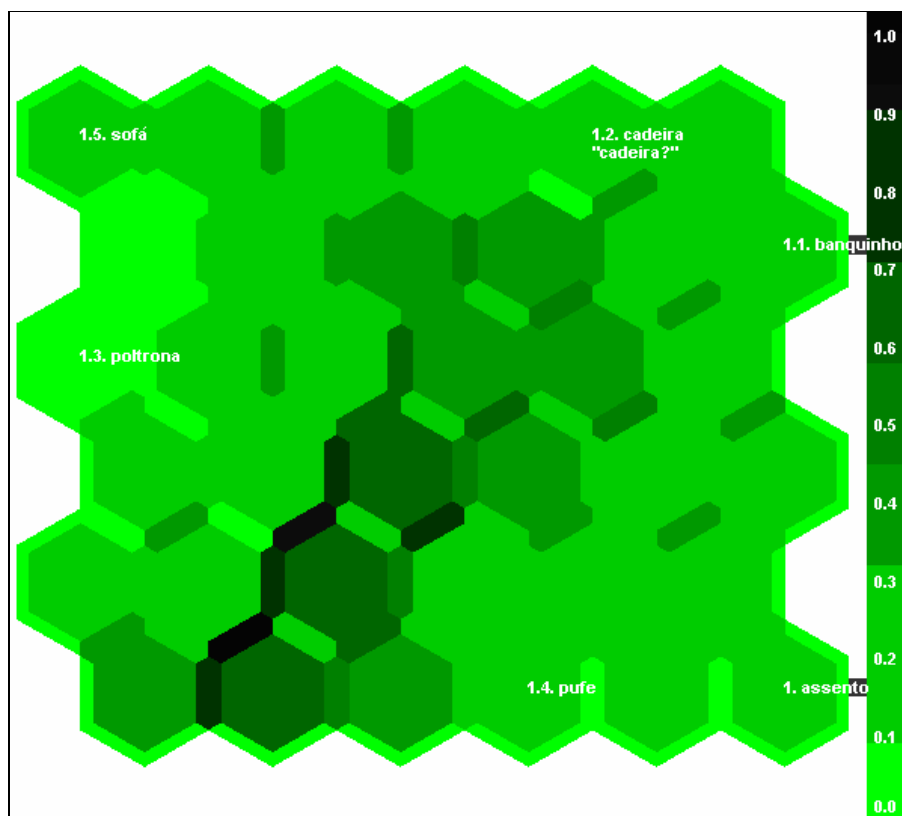


Figura 6.12. – Representação gráfica da rede depois da apresentação do padrão desconhecido

Do ponto de vista lingüístico nós submetemos à rede um conceito e fomos procurar o termo correspondente a esse conceito, realizando um percurso de tipo onomasiológico (cf. p.

73). Do ponto de vista prático, pensem em um usuário de nosso futuro sistema para reconhecimento de texto, que precise de um objeto, /« com encosto »/, /« com pé(s) »/, /« para 1 pessoa »/, /« para se sentar »/, /« com material rígido »/ e que não saiba que esse objeto se chame cadeira: colocando as especificações (os traços semânticos) o sistema retornará ao usuário o termo cadeira.

Depois de ter testado a rede com um padrão desconhecido, agora apresentaremos à rede alguns padrões corrompidos de “cadeiras”, para ver como a rede os classificará.

Submeteremos agora à rede alguns padrões corrompidos (ou incorretos), com alterações no parâmetro 6, como mostrado na Tabela 6.4, para ver como a rede os classificará:

Tabela 6.4. – Vetores de teste de padrões corrompidos no parâmetro S ₆						
	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆
	/« com encosto »/	/« com pé(s) »/	/« para 1 pessoa »/	/« para se sentar »/	/« com braços »/	/« com material rígido »/
“cadeira ?”	1.0	1.0	1.0	1.0	0.0	0.9
“cadeira ??”	1.0	1.0	1.0	1.0	0.0	0.8
“cadeira ???”	1.0	1.0	1.0	1.0	0.0	0.7
“cadeira ????”	1.0	1.0	1.0	1.0	0.0	0.69
“cadeira ?????”	1.0	1.0	1.0	1.0	0.0	0.6
“cadeira ??????”	1.0	1.0	1.0	1.0	0.0	0.5
“cadeira ??????”	1.0	1.0	1.0	1.0	0.0	0.3
“cadeira ??????”	1.0	1.0	1.0	1.0	0.0	0.1
“cadeira ??????”	1.0	1.0	1.0	1.0	0.0	0.0

A Figura 6.13 mostra que a rede reconheceu os primeiros quatro padrões corrompidos como “cadeiras”. O quarto padrão tinha um valor 39% mais baixo do valor de referência. A partir do quinto padrão, com valor 0,6 a rede começa a classificar os padrões em outros neurônios. Contudo, tendo em vista que o parâmetro 6 /« com material rígido »/ é comum a quatro de nossas palavras (“cadeira”, “banquinho”, “poltrona”, “sofá”), é interessante observar o que acontece quando esse parâmetro é muito corrompido. Teoricamente perdendo o parâmetro 6 a rede deveria começar a colocar os vetores corrompidos, gradativamente, mais perto de “pufe”, que não contém esse parâmetro, e de fato, como podemos ver na Figura 6.13, a rede se comporta como esperado.

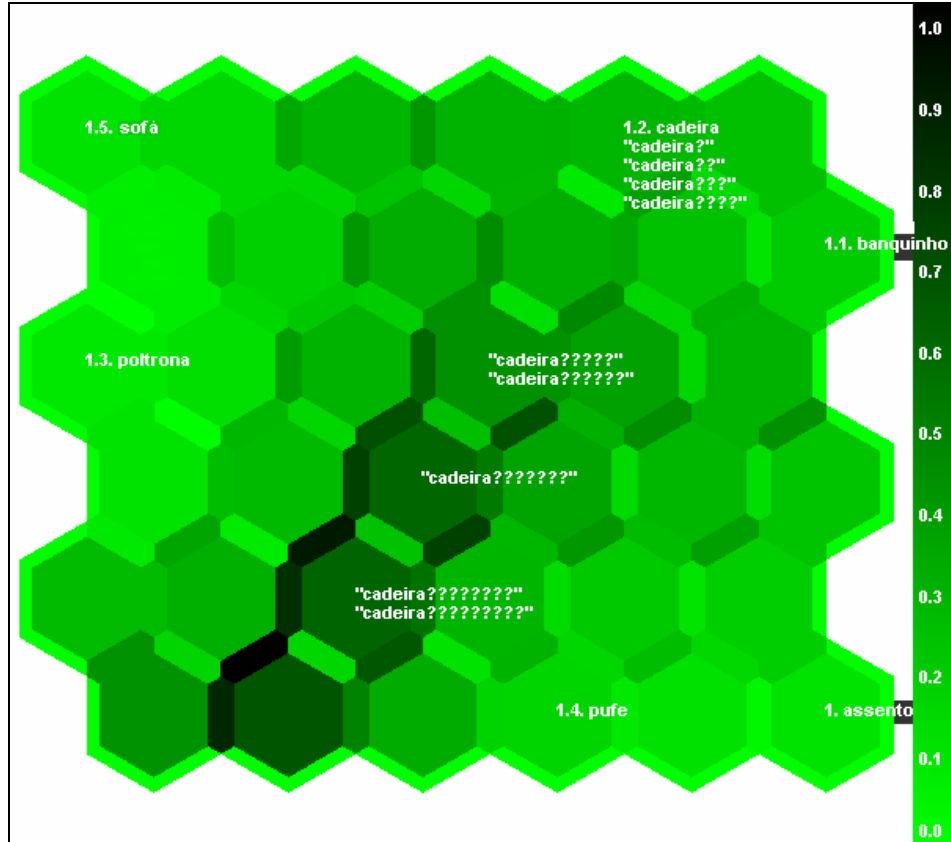


Figura 6.13. – Representação gráfica da rede depois da apresentação de padrões corrompidos no parâmetro 6.

Depois de ter apresentado a rede vetores de teste contendo um único parâmetro corrompido, o parâmetro 6, agora, apresentamos à rede uma série de vetores corrompidos em todos os parâmetros para ver o que acontece. A **Tabela 6.5** mostra esses vetores.

Tabela 6.5 – Vetor de teste com padrões corrompidos em todos os parâmetros						
	S₁	S₂	S₃	S₄	S₅	S₆
	/« com encosto »/	/« com pé(s) »/	/« para 1 pessoa »/	/« para se sentar »/	/« com braços »/	/« com material rígido »/
“cadeira ?”	0.9	0.9	0.9	0.9	0.0	0.9
“cadeira ??”	0.85	0.85	0.85	0.85	0.0	0.85
“cadeira ???”	0.83	0.83	0.83	0.83	0.0	0.83
“cadeira ????”	0.81	0.81	0.81	0.81	0.0	0.81
“cadeira ?????”	0.8	0.8	0.8	0.8	0.0	0.8
“cadeira ??????”	0.7	0.7	0.7	0.7	0.0	0.7
“cadeira ???????”	0.5	0.5	0.5	0.5	0.0	0.5
“cadeira ????????”	0.3	0.3	0.3	0.3	0.0	0.3
“cadeira ?????????”	0.1	0.1	0.1	0.1	0.0	0.1

Como podemos ver na Figura 6.14, quando todos os padrões do vetor são corrompidos, a rede continua sendo muito eficaz, classificando corretamente os padrões apresentados, com valores de até 19% inferiores aos valores de referências. Os vetores com valores mais corrompidos tendem a ficar, gradativamente, mais perto de “pufe”, parecendo determinante a diminuição do valor do s_6 /« com material rígido »/ e em seguida de “assento”. O último vetor apresenta valores de 90% inferiores aos valores de referência do vetor de “cadeira” e é classificado no neurônio de “assento”. Do ponto de vista semântico isso é pertinente tendo em vista o fato de que assento é o termo hiperônimo da série, em outras palavras o menos marcado – contém somente dois traços semânticos.

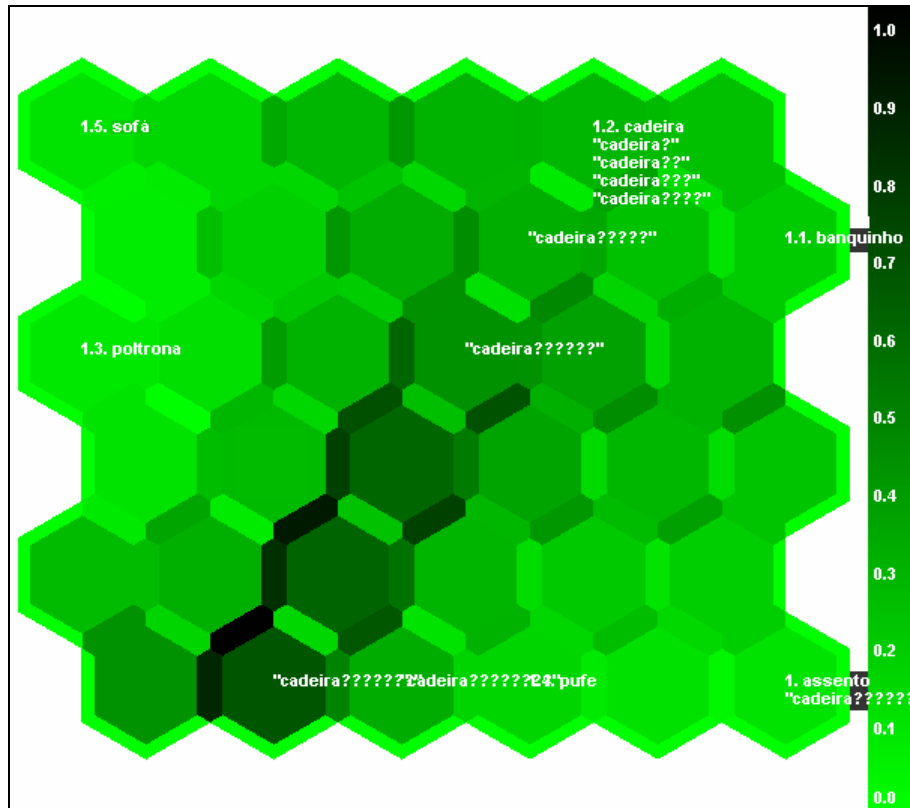


Figura 6.14 – Representação gráfica da rede depois da apresentação de padrões corrompidos em todos os parâmetros.

Como último teste, pegamos uma série de vetores com o parâmetro s_1 (/ \ll com encosto \gg) corrompido. A diferença entre um banquinho e uma cadeira é dada pelo parâmetro 1, “cadeira” contém esse parâmetro, enquanto “banquinho” não. A Tabela 6.6 mostra os vetores contendo o parâmetro 1 corrompido.

Tabela 6.6 – Vetor de teste com padrões muito corrompidos no parâmetro S_6						
	S_1	S_2	S_3	S_4	S_5	S_6
	/« com encosto »/	/« com pé(s) »/	/« para 1 pessoa »/	/« para se sentar »/	/« com braços »/	/« com material rígido »/
“cad-banquinho?”	0.9	1.0	1.0	1.0	0.0	1.0
“cad-banquinho??”	0.8	1.0	1.0	1.0	0.0	1.0
“cad-banquinho???”	0.7	1.0	1.0	1.0	0.0	1.0
“cad-banquinho????”	0.6	1.0	1.0	1.0	0.0	1.0
“cad-banquinho?????”	0.5	1.0	1.0	1.0	0.0	1.0
“cad-banquinho??????”	0.3	1.0	1.0	1.0	0.0	1.0
“cad-banquinho???????”	0.1	1.0	1.0	1.0	0.0	1.0
“cad-banquinho????????”	0	1.0	1.0	1.0	0.0	1.0

A Figura 6.15 mostra a rede depois da submissão dos vetores contidos na Tabela 6.6. Os primeiro quatro vetores foram reconhecidos como “cadeiras”. O último dos quatro tem um valor de 40% mais baixo do valor de referência. Acima do valor de 0,6 a rede não classifica mais os vetores como “cadeiras”. É interessante observar que somente o último vetor, contendo o parâmetro 1 igual a zero, é reconhecido como “banquinho”, os demais são classificados à direita do neurônio de “cadeira” e acima do de “banquinho”. Com efeito, um banquinho não pode ter, para ser considerado tal, conforme a descrição semântica dada na **Tabela 6.1** nenhum encosto, caso contrário, ele seria considerado uma cadeira.

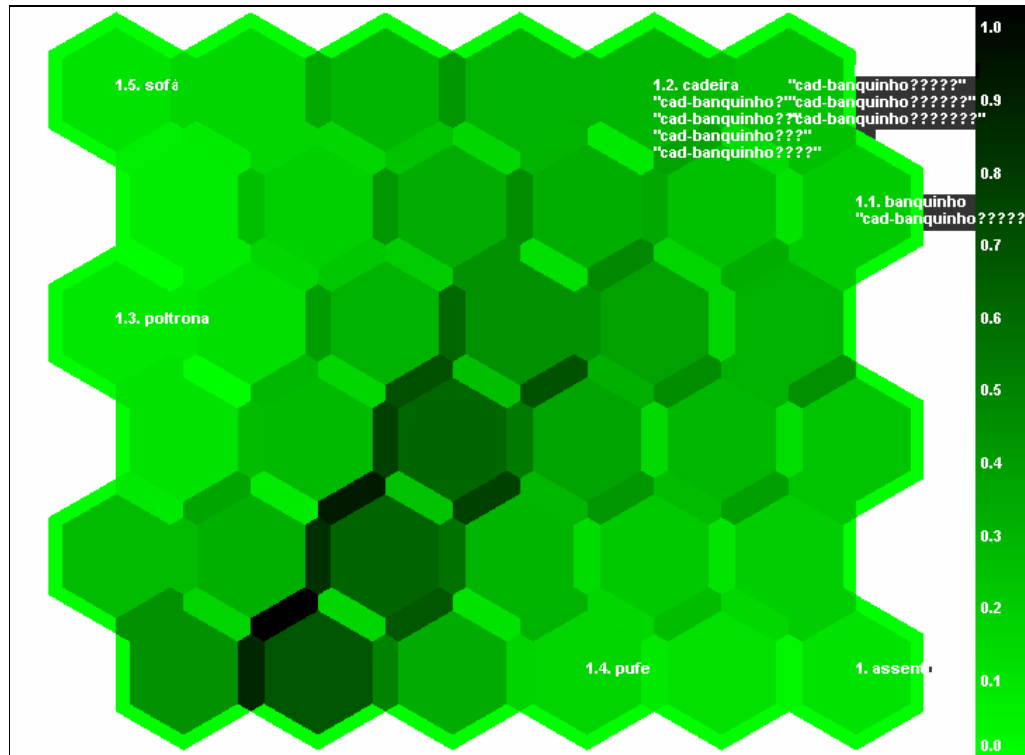


Figura 6.15 – Representação gráfica da rede depois da apresentação de padrões corrompidos no parâmetro 1

Os testes apresentados são somente alguns dos tantos que efetuamos. Eles deram bons resultados, confirmando as qualidades da rede e sua aptidão para reconhecer padrões de tipo lexical. Para testar, ulteriormente, a validade do modelo proposto seriam necessários mais testes e trabalhar com conjuntos de termos maiores, o que pretendemos fazer, continuando, futuramente, nossos estudos sobre as Redes Neurais Artificiais.

Conclusões

Objetivo desse trabalho era estudar os mecanismos que permitem utilizar uma rede neural artificial visando interpretar textos, tendo em vista que esse trabalho deveria ser utilizado, futuramente, para criar uma interface, em um ambiente de co-projeto, capaz de agrupar/classificar termos/conceitos, reconhecendo padrões de tipo textual.

Para alcançar nossos objetivos de pesquisa estudamos, no Capítulo 1, as características principais das RNAS (componentes, funções de ativação e tipos de arquiteturas).

Em seguida, no Capítulo 2, estudamos de maneira mais aprofundada a rede de Kohonen, rede que escolhemos para nosso trabalho. A rede de Kohonen apresenta características e funcionamento bem diferentes das demais Redes Neurais Artificiais: ela é constituída por uma única camada de Neurônios que ao mesmo tempo servem de entrada e saída para o sinal e três processos diferentes (competitivo, cooperativo e adaptativo sináptico) regem seu funcionamento.

No Capítulo 3, um breve histórico da Inteligência Artificial e a definição de alguns dos conceitos-chave permitiram-nos precisar o objeto de nossa pesquisa do ponto de vista teórica e nos forneceu alguns elementos importantes sobre os diferentes tipos de aprendizado.

O estudo sobre os modelos booleano e de espaço vetorial, no Capítulo 4, foi de fundamental importância para entender como transformar textos em vetores, e para compreender como comparar e medir a similaridade entre vetores.

No Capítulo 5, analisamos o modelo semântico de Bernard Pottier, partindo das definições de percurso semasiológico e de percurso onomasiológico, que nos permitiram melhor entender como o lingüista explica os processos de decodificação e decodificação da mensagem. Em seguida estudamos o significado de sema, conceito, e noema, que são os elementos constitutivos do significado de uma unidade lexical. O exemplo clássico de análise

semântica proposto por ele de *cadeira, poltrona, banquinho, sofá, pufe e assento* nos serviu de base para a futura implementação da rede de Kohonen.

No Capítulo 6 implementamos uma Rede de Kohonen composta de trinta e seis neurônios (6 x 6) por meio do *software* Nenet 1.1. do *Nenet Team* da Universidade de Tecnologia de Helsinski. A escolha desse *software* deve-se ao fato de que dispõe de uma interface gráfica bem feita e reproduz com fidelidade o modelo de rede proposto por Kohonen. Começamos a implementação da Rede com a transformação em vetores dos termos analisados por Pottier, isto é *cadeira, poltrona, banquinho, sofá, pufe e assento*. Depois de termos inicializado e treinado a rede, submetemos os vetores de teste contendo as etiquetas correspondentes aos termos que representavam. A rede mostrou reconhecer os padrões submetidos agrupando os mesmos em função das características semânticas dos termos correspondentes.

A rede, por meio de representações gráficas policromáticas, permite mostrar de maneira aprofundada as relações semânticas que existem entre os diferentes termos.

Em seguida submetemos à rede um padrão desconhecido para testar sua capacidade de classificar novos padrões, tendo resultados positivos.

Como último teste apresentamos à rede padrões corrompidos, que foram também classificados corretamente.

Como nos ensinam os epistemólogos modernos, em particular Karl Popper e Thomas Kuhn, a ciência moderna não é somente indutiva, mas também dedutiva e o fato de nosso sistema parecer funcionar não é condição suficiente para que ele seja futuramente adotado e utilizado. Caberá à República dos cientistas, em primeira instância, diz Thomas Kuhn, aprovar ou não nosso trabalho. Contudo, dentre os critérios sugeridos, por Thomas Kuhn para avaliar um sistema (um novo paradigma) há sua relativa simplicidade e coesão interna, e a capacidade de prever novos fenômenos.

Nesse sentido, acreditamos que nosso trabalho possa ser utilizado futuramente para finalidades diferentes daquelas as quais nos propusemos. Mais especificamente, nós pensamos em sistemas para reconhecimento de objetos. Seria possível reconhecer objetos partindo de traços semânticos separados, como supostamente faria o cérebro humano? Ou seja, seria possível que uma máquina reconheça uma cadeira a partir das diferentes partes que a compõem, isto é, encosto e pés, por exemplo? E ainda, seria possível distinguir uma cadeira de um banquinho, partindo do reconhecimento desses traços semânticos como proposto neste estudo?

Esperamos que nosso trabalho possa estimular novos estudos nesse sentido, promovendo e melhorando os conhecimentos nas Áreas da Inteligência Artificial, da Automação e do Processamento da Linguagem Natural.

Referências bibliográficas

ACKERMAN, Rich. **Vector Model Information Retrieval.**

<http://www.hray.com/5264/math.htm>. Acesso: em junho 2006.

BAEZA-YATES, Ricardo; RIBEIRO-NETO, Berthier. **Modern information retrieval.** New York: ACM Press, 1999.

BRAGA, Antônio de Pádua; CARVALHO (de), André Ponce de Leon F.; LUDEMIR, Teresa Bernarda. **Fundamentos de Redes Neurais Artificiais.** 11º Escola de Computação. Rio de Janeiro: UFRJ, 1998.

HAYKIN, Simon. **Redes Neurais: Princípios e Prática.** Bookman, 2001.

HEBB, Donald Olding. **The organization of behavior.** New York: Wiley, 1949.

HOPFIELD, John J. Neural network and physical system with emergent collective properties. **Proc. Nat. Acad. Sci.**, 79:2554-8, 1982.

HUMBOLDT (von), Karl Wilhelm. **La diversità delle lingue.** Roma: Laterza, 2000.

KAESTNER, Celso A. A.. **Tratamento Automático de Documentos em Texto Livre.** Documento em PDF. Site: www.ppgia.pucpr.br/~kaestner/ir/2-CONGED.ppt. Acesso em junho 2006.

KOHONEN, Teuvo. An adaptive associative memory principle. **IEEE Transactions on Computers**, C-23:444-445, 1974.

_____. **Self-organization and associative memory.** Berlin-Heidelberg: Springer-Verlag, 1989.

_____. Self-organized formation of topologically correct feature maps. **Biological Cybernetics**, 43, 1982.

LEVINE, M.; SHEFNER, J. M. **Fundamentals of Sensation and Perception.** 2a ed. Pacific Grove, CA: Books/Cole, 1991.

MARQUES de SÁ, Joaquim P. **Reconhecimento de padrões**. Site do Prof. Dr. Joaquim P. Marques de SÁ (Prof. Associado), da Faculdade de Engenharia da Universidade do Porto. (<http://paginas.fe.up.pt/~jmsa/recpad/>). Acesso em: Março 2006.

MARRANGHELLO, Norian. **Redes de Petri aplicadas à Síntese de Sistemas Digitais**. Apostila do curso de Redes de Petri, oferecido no curso de Pós-Graduação de Engenharia Elétrica da Unesp – Campus de Ilha Solteira, 2005.

McCULLOCH, Warren; PITTS, Walter. A Logical Calculus of the Ideas Immanent In Nervous Activity. **Bulletin of Mathematical Biophysics**. n. 5, p. 115-133, 1943.

MINUSSI, Carlos Roberto. **Redes Neurais**: Introdução e principais conceitos. Apostila do Curso de Neurais Artificiais, oferecido no curso de Pós-Graduação de Engenharia Elétrica da Unesp – Campus de Ilha Solteira, 2005.

NENET TEAM. **Nenet**: demo version 1.1. [S.l.]: Nenet Team, 1999. Site <http://koti.mbnet.fi/~phodju/nenet/Nenet/General.html>. Acesso em: janeiro 2006.

POTTIER, Bernard. **Linguistique générale**: théorie et description. Paris : Klincksieck, 1985.

_____. **Sémantique générale**. Paris : PUF, 1992.

_____. **Théorie et analyse en linguistique**. 2^e éd. Paris : Hachette, 1987.

_____. Vers une sémantique moderne. **Travaux de linguistique et de littérature**. vol. 2. n^o 1. Strasbourg : Klincksieck, p. 107-137. 1964.

RAUBER, Thomas Walter. **Reconhecimento de Padrões. Mini-Curso**. JAI '97 – XVI. JOURNEY OF ACTUALIZATION IN COMPUTER SCIENCE XVII. CONGRESS OF THE BRAZILIAN COMPUTER SCIENCE SOCIETY Brasília, DF, Brazil, 1997. (Arquivo PDF, Disponível em <http://www.inf.ufes.br/~thomas/home/courses.html>). Acesso em: Março 2006.

SALTON, Gerard; BUCKLEY, C. Term-weighting approaches in automatic retrieval. **Information Processing & Management**. n. 24(5), 513-523, 1988.

SALTON, Gerard; WONG, A.; YANG, C. S. Vector Space Model for Automatic Indexing. **Communication of the ACM**. 18(11), 613-620, 1975.

SNNS-Group. **Stuttgart Neural Network Simulator 4.2**. Stugart – Tübingen: Institute for Parallel and Distributed High-Performance Systems (IPVR), University of Stuttgart – Wilhelm-Schickard-Institut für Informatik (WSI), University of Tübingen, 1998. Site <http://www-ra.informatik.uni-tuebingen.de/SNNS/>. Acesso em: abril 2006.

WANGENHEIM (von), Aldo. **Reconhecimento de Padrões**. Apostila do Prof. Dr. Aldo von Wangenheim (UFSC). Site <http://www.inf.ufsc.br/~awangenh/RP/programa.html>. Acesso em: abril 2006.

WASSERMAN, P. D. **Neural Computing: Theory and Practice**. New York: Van Nostrand Reinhold.

Wikipedia. <http://pt.wikipedia.org/wiki/>. Acesso em maio 2006.