

FRANCISCO JAVIER ROPERO PELÁEZ

REDES NEURAIS E LÓGICA FUZZY  
SOB A PERSPECTIVA DE UMA TEORIA  
ALGÉBRICA DA PROBABILIDADE

Tese apresentada à Escola  
Politécnica da Universidade de  
São Paulo para obtenção do  
título de Doutor em Engenharia

São Paulo

2001

CONSULTA

FT-1641

FRANCISCO JAVIER ROPERO PELÁEZ

**REDES NEURAIS E LÓGICA FUZZY  
SOB A PERSPECTIVA DE UMA TEORIA  
ALGÉBRICA DA PROBABILIDADE**

Tese apresentada à Escola  
Politécnica da Universidade de São  
Paulo para obtenção do título de  
Doutor em Engenharia

Área de Concentração:  
Engenharia Mecânica / Mecatrônica

Orientador:  
MARCELO GODOY SIMÕES

Co-orientador:  
Newton Maruyama

São Paulo

2001

“El hombre es, si lo desea, el escultor de su propio cerebro”.

(Santiago Ramón y Cajal)

A David, com carinho e admiração.

## AGRADECIMENTOS

Quero agradecer a Deus pelo fato de ter colocado de maneira providencial em minha vida as pessoas, os acontecimentos e as inspirações que me ajudaram ao desenvolvimento das pesquisas desta tese.

Deste modo, quero primeiro agradecer a Deus pelos meus pais, que foram aqueles que prepararam a terra durante anos para eu pudesse pesquisar.

Agradeço a Deus por ter conhecido no Japão os professores Sun-Ichi Amari e Kazu Aihara, que ajudaram a colocar a semente nessa terra. Bem como, minha esposa Shirley. Ela foi o sol que fez germinar a semente. No Japão também conheci os Professores Jun Okamoto e Celso Masatoshi Furukawa. Neste país múltiplas coincidências me levaram a conhecer meu orientador Professor Marcelo Godoy Simões.

Obrigado Deus por aquela terra fértil onde Você colocou a semente. Uma vez a semente virou uma pequena planta, esta planta foi transplantada às terras fortes e férteis do Brasil, onde o meu orientador Professor Godoy e outras pessoas como o meu co-orientador Professor Newton Maruyama e os meus companheiros Silvio, Petrônio, Nilson, Ettore, Alberto, Johnny e Paulo, trataram-na com paciência e dedicação. Agradeço à FAPESP pela bolsa de pesquisa. Obrigado Brasil.

Agradeço a Deus por ter-me permitido conhecer a finalidade de minhas pesquisas com o nascimento do meu filho David.

## *RESUMO*

Nesta tese foi desenvolvida uma nova teoria Euclidiana da Probabilidade que permite visualizar os eventos estatísticos como vetores. Os conceitos de ângulo e projeção permitiram desenvolver um novo tipo de algoritmo de Gram-Schmidt para encontrar uma base de vetores ortogonais a partir de outros vetores quaisquer. Usando-se na base os eixos que sejam ao mesmo tempo os chamados componentes principais, o número total de eixos pode ser reduzido. Assim, um novo algoritmo de obtenção de componentes principais foi desenvolvido. Estes fundamentos matemáticos serviram para embasar de maneira diferente a lógica *fuzzy* e as redes neurais artificiais. Na área da lógica *fuzzy* estas equações fornecem um método diferente de desenhar analiticamente as funções de pertinência e permitem encontrar as regras composicionais de inferência. Na área das redes neurais, permitiram o desenho e o fácil entendimento de uma nova rede neural baseada na neuro-fisiologia do tálamo, que obtem os componentes principais para lograr uma compressão eficiente da informação. Este tálamo artificial foi implementado em Matlab.

## *ABSTRACT*

This thesis develops a new Theory of Probability that allows the understanding of statistical events as vectors in an Euclidean space. The concepts of angle and projection allowed the developing of a new Gram-Schmidt-type algorithm for finding a basis of orthogonal vectors from any other set of vectors. The number of axes in this basis can be reduced when the axes are the so-called Principal Components. A new algorithm for extracting the Principal Components was also developed. These mathematical foundations served for grounding Fuzzy Logic and Neural Networks in a different way. Regarding Fuzzy Logic, these equations represent a different method for analytically designing membership functions and for finding the compositional inference rules. In the Neural Network field these foundations allowed the easy understanding of a new neural network based on the neuro-physiology of thalamus that extracts the Principal Components for achieving an efficient compression of information. This artificial thalamus was implemented in Matlab.

## Erratas:

Peláez, Francisco Javier Ropero  
Redes Neurais e Lógica Fuzzy sob a Perspectiva de  
uma Teoria Algébrica da Probabilidade. São Paulo,  
2001.  
188 p.  
Tese (Doutorado) – Escola Politécnica da  
Universidade de São Paulo. Departamento de  
Engenharia Mecânica.  
I. Estatística Euclidiana. I. Universidade de São  
Paulo. Escola Politécnica. Departamento de  
Engenharia Mecânica. II.t

### **COMPARAÇÃO DA REDE TALÂMICA COM DUAS ARQUITETURAS NEURAI SEMELHANTES: A REDE ART (GROSSBERG, 1987)<sup>1</sup> E A REDE DE OJA-SANGER (SANGER, 1989)<sup>2</sup>**

#### **Comparação com a rede ART:**

##### **Semelhanças:**

- Ambas as redes têm duas camadas: uma camada de entrada e uma camada competitiva.
- Na camada competitiva o neurônio ativo é aquele cujos pesos são mais parecidos com a saída da primeira camada.
- Existem conexões para frente e conexões de retroalimentação entre as duas camadas.

##### **Diferenças:**

- A rede ART faz uma classificação dos padrões apresentados enquanto que a rede talâmica realiza, além disto, uma extração dos componentes principais.
- A retroalimentação é inibitória na rede talâmica sendo excitatória na rede ART.

<sup>1</sup> Grossberg, S. (1987) Adaptive pattern classification and universal recording: parallel development and coding of neural feature detectors. *Biol. Cybern.*, **23**: 187-202

<sup>2</sup> Sanger, T.D. (1989) Optimal unsupervised learning in a single-layer linear feed-forward neural network. *Neural Networks*, **12**:459-473

- Na rede talâmica a retroalimentação inibitória é uma reconstrução do padrão de entrada enquanto que a retroalimentação na rede ART fornece o padrão médio (ou protótipo) de uma determinada classe de padrões
- O processamento na rede talâmica é restrito a cada neurônio enquanto que na rede ART existem valores globais a serem considerados (ganhos G1 e G2) e um fator de vigilância.

### **Comparação com a rede de Oja-Sanger:**

#### Semelhanças:

- Ambas as redes têm duas camadas.
- Ambas as redes extraem as componentes principais de um conjunto de padrões.

#### Diferenças:

- A rede de Oja-Sanger somente tem conexões para frente.
- Cada um dos neurônios da segunda camada na rede de Sanger é ativado numa ordem determinada, enquanto que na rede talâmica, cada neurônio é ativado por um processo competitivo.
- Os termos subtrativos que aparecem no algoritmo de reforço sináptico dos pesos da rede de Sanger equivalem aos valores dos pesos das conexões de retroalimentação inibitórias da rede talâmica.

### **CONCLUSÃO DA PARTE I:**

Nesta primeira parte da tese foi desenvolvida uma nova teoria da probabilidade onde os eventos estatísticos são considerados vetores num espaço Euclidiano.

Nesta nova teoria, a probabilidade de um evento calcula-se como a projeção desse evento/vetor sobre um evento/vetor específico que é o universo normalizado. Esses eventos/vetores têm grandes semelhanças com os chamados conjuntos “fuzzy” pois estão formados pela soma ponderada de componentes que podem ser de natureza qualitativa. Semelhante ao que acontece com os chamados conjuntos “fuzzy”, nesta nova teoria da probabilidade a interseção de eventos complementares não é zero.

É possível obter-se uma base de vetores ortogonais a partir de um conjunto de vetores não ortogonais aplicando-se o algoritmo de Gram-Schmidt. Desenvolveu-se uma alteração do algoritmo para a obtenção dos componentes principais de um conjunto de padrões.

## ADENDA À CONCLUSÃO DA PARTE III.

**Oferecemos aqui uma síntese do algoritmo da rede talâmica:**

1. Um padrão é fornecido na entrada da rede. As inibições de retroalimentação são zeradas. Todos os neurônios que estavam em período refratário<sup>3</sup> voltam ao normal (ver passo 4.2).
2. Na camada 1 calcula-se o vetor diferencia entre o padrão de entrada e o total das inibições de retroalimentação.
3. Uma função logística é aplicada a cada componente deste vetor fornecendo um valor binário. O resultado multiplica-se escalarmente pelos pesos de cada neurônio da segunda camada, obtendo-se os potenciais pós-sinápticos dos neurônios da camada 2.
- 4.1. Se não existe nenhum neurônio disparando na segunda camada, calcula-se o neurônio vencedor (que vai disparar) na 2ª camada. Este neurônio é o que tem potencial pós-sináptico maior.
- 4.2. Se existe algum neurônio disparando na segunda camada, este neurônio continuará disparando até que a primeira derivada do seu potencial pós-sináptico seja inferior a um determinado valor negativo (No programa da pag. 158 este valor é  $-0.01$ ). Quando isto acontece o neurônio entra em período refratário. Isto significa que este neurônio não vai poder disparar no futuro.
5. Multiplicam-se os valores binários dos disparos do neurônio ganhador da segunda camada pelos pesos de retroalimentação desde o neurônio vencedor aos neurônios da primeira camada. O resultado é um vetor de inibições desde a 2ª para a 1ª camada.
6. Este vetor de inibições que é ponderado por um fator menor do que 1 (tipicamente 0.94) vai ser somado as inibições produzidas pelos neurônios ganhadores em anteriores iterações obtendo-se a retroalimentação inibitória total.
7. Repetem-se os passos 2 até 6 um número determinado de vezes (no exemplo das paginas 126-136) este número era de vezes era 40.
8. Retorno ao passo 1 após o número estabelecido de iterações.

### **Notas:**

Cada neurônio dispara um valor binário. O valor do potencial pós-sináptico é a probabilidade do neurônio fornecer um 1 na sua saída.

Em cada momento os pesos (para frente e de retroalimentação) são modificados através de uma probabilidade condicional do tipo  $P(B/A)$  onde B é o evento pós-sináptico e A o evento pré-sináptico.

Nos pesos que vão da 1ª em direção a 2ª camada, o evento pré-sináptico A é o valor da ativação do neurônio da primeira camada no caso deste neurônio receber também uma inibição de um valor mínimo pré-determinado. O evento pós-sináptico B é o valor da ativação do neurônio da 2ª camada.

---

<sup>3</sup> No período refratário os neurônios biológicos ficam temporariamente inativos depois de cada disparo.

Nos pesos que vão da 2<sup>a</sup> em direção a 1<sup>a</sup> camada o evento pós-sináptico B é o valor do padrão de entrada no neurônio da 1<sup>a</sup> camada enquanto que o evento pré-sináptico é o valor da ativação do neurônio na 2<sup>a</sup> camada.

### **Parâmetros normalmente usados:**

São usados inicialmente pesos aleatórios de valor pequeno (pag. 149). O parâmetro “concentração de neuro-transmissor” permite mudar a rapidez de ajuste dos pesos em cada iteração do algoritmo. Este parâmetro foi ajustado no programa para o valor 50, significando que o calculo dos pesos foi repetido 50 vezes para cada iteração (passos 2 até 6). O fator usado na etapa 6 é chamado “fator de inibição” e tem um valor entre 0.92 e 1, tipicamente 0.96

### **CONCLUSÃO FINAL**

Desde que Pitágoras mostrou a relação estreita entre a matemática e a percepção musical, o ser humano teve a suspeita de que, no cérebro, deveriam acontecer processos de índole matemática. Nesta tese, a matemática desenvolvida na primeira parte serviu para demonstrar uma nova teoria de processamento da informação sensorial a nível do tálamo cerebral. Deste modo, desenvolveu-se uma nova rede neural que produz uma extração de atributos a partir do conjunto de padrões mostrados na entrada. Postula-se que estes atributos sejam as chamadas componentes principais do conjunto de padrões. Esta teoria matemática serve também para embasar de uma forma diferente a lógica “fuzzy” permitindo o desenho analítico das funções de pertinência e as relações “fuzzy”.

---

### **ALTERAÇÕES NAS REFERÊNCIAS.**

[31] Ropero Peláez, J. (2000) Towards a neural network-based therapy for hallucinatory disorders. *Neural Networks. 2000 Special Issue* **13** (2000): 1047-1061.

Na pag. 8 mudar a referência 38 pela 44.

Na pag. 78, segunda linha, inserir: [16] e [22] após o texto: “...segundos recentes dados neurofisiológicos”.

---

### **LISTA DE FIGURAS:**

- |  |    |
|--|----|
| <b>Figura 1</b> Neste exemplo, um simples universo é reduzido a "átomos".  | 6  |
| <b>Figura 2</b> Cada categoria de “átomos”, é classificada de acordo com o objeto a que eles pertencem.  | 7  |
| <b>Figura 3</b> Os objetos podem ser identificados não somente através da característica físicas dos “átomos” que os constituem, mas também por características subjetivas.. | 11 |
| <b>Figura 4</b> Projetando-se qualquer vetor com a ponta da flecha no plano P sobre u, obtém-se a probabilidade de B.  | 45 |

|                  |  |     |
|------------------|--|-----|
| <b>Figura 5</b>  | Nesta figura são mostradas diferentes formas de se obter um conjunto de funções de pertinência a partir de várias trajetórias desenhadas no plano de equiprobabilidade P.  | 51  |
| <b>Figura 6</b>  | Forma de uma função de pertinência a partir da trajetória desenhada no plano P.  | 53  |
| <b>Figura 7</b>  | Funções de pertinência desenhadas para uma situação na qual as únicas informações disponíveis são (a) que o domínio da variável tamanho vai, por exemplo, de 2 à 12, e (b) o número de medidas pertencentes as três diferentes categorias são $n(\text{grande})=15$ , $n(\text{Média})=29$ e $n(\text{Pequena})=9.5$ . Usando-se a informação geométrica dada pela fig. 6.a $t1-2$ e $t1-2$ são obtidos: $t1-2=4.04$ e $t2-3=5.96$ .                                       | 56  |
| <b>Figura 8</b>  | É mostrado o mesmo caso da fig.7 mas acrescentando-se duas novas categorias vazias: a categoria muito pequena e a categoria muito grande. Estas novas categorias produzem os novos componentes laterais horizontais desta figura.  | 58  |
| <b>Figura 9</b>  | Desenho de uma função trapezoidal.   | 59  |
| <b>Figura 10</b> | É possível considerar-se as funções de pertinência como uma função de decisão de Bayes.  | 62  |
| <b>Figura 11</b> | Quando a cardinalidade das medidas "muito grande", $n(\text{muito grande})$ diminui, o limiar $x$ é deslocado para à direita, significando que um alto valor de $t$ é necessário para classificar $B(t)$ como "muito grande". Quando $n(\text{"muito\_grande"})=0$ , então a distância $t$ e o limiar $x$ são infinitos. A função de pertinência etiquetada como "grande" torna-se uma função trapezoidal, enquanto que a função de pertinência "muito grande" é achatada. | 63  |
| <b>Figura 12</b> | Representação 3-D da regra: Se A é médio e B é grande, então aplique uma potência média. A pirâmide representa a combinação de funções de pertinência "Médio A", "Grande B" e "Média potência".  | 65  |
| <b>Figura 13</b> | A conjunção "e" de duas funções de pertinência triangulares pode ser realizada de duas maneiras: (a) computando-se o mínimo, na qual linhas de nível quadradas são obtidas, ou (b) computando-se a multiplicação de ambas as funções, o que leva a um leve arredondamento da pirâmide. Isto corresponde ao cálculo da interseção de duas funções de pertinência estatisticamente independentes.  | 67  |
| <b>Figura 14</b> | Dois redes neurais encadeadas para classificação e integração.   | 79  |
| <b>Figura 15</b> | Estrutura e conexões do tálamo e da camada IV do córtex.   | 84  |
| <b>Figura 16</b> | Exemplo de processamento de um padrão no tálamo  | 95  |
| <b>Figura 17</b> | A arquitetura do tálamo é a mesma que poderia ser obtida dobrando-se uma rede auto-associativa.  | 100 |
| <b>Figura 18</b> | Compressão de dados para serem enviados através de um canal de comunicações.   | 101 |
| <b>Figura 19</b> | Os componentes do auto-vetor que multiplicado escalarmente pelo vetor X gera um dos componentes principais, poderiam ser consideradas como pesos de uma rede neural que relaciona uma entrada (neste caso de 14 componentes) com uma saída (neste caso de quatro componentes).   | 103 |
| <b>Figura 20</b> | Esquema de conexões entre os neurônios tálamo-corticais e os neurônios reticulares.  | 113 |
| <b>Figura 21</b> | Exemplo de funcionamento.  | 126 |
| <b>Figura 22</b> | Exemplo de funcionamento.  | 127 |
| <b>Figura 23</b> | Exemplo de funcionamento.  | 128 |
| <b>Figura 24</b> | Exemplo de funcionamento.  | 129 |
| <b>Figura 25</b> | Exemplo de funcionamento.  | 130 |
| <b>Figura 26</b> | Exemplo de funcionamento.  | 131 |
| <b>Figura 27</b> | Exemplo de funcionamento.  | 132 |
| <b>Figura 28</b> | Exemplo de funcionamento.  | 133 |
| <b>Figura 29</b> | Exemplo de funcionamento.  | 134 |
| <b>Figura 30</b> | Exemplo de funcionamento.  | 135 |
| <b>Figura 31</b> | Exemplo de funcionamento.  | 135 |
| <b>Figura 32</b> | Exemplo de funcionamento.  | 136 |
| <b>Figura 33</b> | Exemplo de funcionamento.  | 138 |

|  |     |
|--|-----|
| <b>Figura 34</b> Exemplo de funcionamento. | 139 |
| <b>Figura 35</b> Exemplo de funcionamento. | 140 |
| <b>Figura 36</b> Exemplo de funcionamento. | 141 |
| <b>Figura 37</b> Exemplo de funcionamento. | 141 |
| <b>Figura 38</b> Exemplo de funcionamento. | 145 |
| <b>Figura 39</b> Exemplo de funcionamento. | 145 |
| <b>Figura 40</b> Exemplo de funcionamento. | 146 |
| <b>Figura 41</b> Exemplo de funcionamento. | 146 |
| <b>Figura 42</b> Exemplo de funcionamento. | 147 |
| <b>Figura 43</b> Exemplo de funcionamento. | 147 |
| <b>Figura 44</b> Exemplo de funcionamento. | 148 |
| <b>Figura 45</b> Exemplo de funcionamento. | 148 |

## SUMÁRIO:

|  |            |
|--|------------|
| Introdução:  | 3          |
| <b>PARTE 1. NOVOS EMBASAMENTOS DA ESTATÍSTICA SOB O PONTO DE VISTA DA ÁLGEBRA LINEAR.</b>  | <b>6</b>   |
| Eventos estatísticos considerados como vetores num espaço Euclidiano.  | 6          |
| Probabilidade como uma projeção.   | 14         |
| Probabilidade da interseção de eventos.  | 19         |
| Um enfoque algébrico para a regra de decisão de Bayes  | 23         |
| Eventos estatísticos independentes   | 26         |
| Transformações lineares em espaços Euclidianos probabilísticos.  | 30         |
| Uma versão estatística do algoritmo de Gram-Schmidt.   | 33         |
| Uma alternativa ao algoritmo de ortogonalização de Gram-Schmidt a fim de permitir a extração dos chamados componentes principais do conjunto de padrões. | 36         |
| Comparação entre diferentes interpretações geométricas de probabilidade.   | 39         |
| <b>PARTE II. A LÓGICA FUZZY SOB A PERSPECTIVA DA ESTATÍSTICA EUCLIDIANA</b>  | <b>42</b>  |
| Introdução   | 42         |
| Funções de pertinência obtidas a partir de trajetórias em um plano.  | 45         |
| Maximização da entropia e funções de pertinência.  | 48         |
| Entropia e desenho de funções de pertinência.  | 53         |
| Funções de pertinência e lei de decisão de Bayes   | 62         |
| Transformação linear e a regra composicional da inferência.  | 65         |
| Conclusões da Parte II   | 72         |
| <b>PARTE III APLICAÇÃO DA ESTATÍSTICA EUCLIDIANA NO DESENVOLVIMENTO DE UMA NOVA REDE NEURAL (REDE TALÂMICA)</b>  | <b>75</b>  |
| Revisão histórica.   | 77         |
| Localização e estrutura do tálamo  | 83         |
| Modelagem dos pesos sinápticos   | 86         |
| Propriedades dos neurônios da camada reticular   | 88         |
| Propriedades dos neurônios da camada tálamo-cortical.  | 90         |
| Computação de Gram-Schmidt no modelo do tálamo.  | 91         |
| Introdução: A rede talâmica: uma rede neural de extração de componentes principais   | 98         |
| Redes auto-associativas, uma primeira aproximação para entender a extração de componentes principais na rede talâmica.                                   | 99         |
| Componentes principais para otimizar a compressão de dados   | 101        |
| Condições para que os pesos evoluam na direção dos componentes principais  | 103        |
| Reforço hebbiano e componentes principais  | 105        |
| Competitividade e recorrência para obter componentes principais distintos.   | 116        |
| Demonstração do PCA no tálamo usando-se a estatística Euclidiana.  | 118        |
| Conclusões da Parte III  | 123        |
| <b>PARTE IV. METODOLOGIA E IMPLEMENTAÇÃO</b>   | <b>124</b> |
| <b>DEMONSTRAÇÃO DA CAPACIDADE DE RECONSTRUÇÃO DE PADRÕES DA REDE TALÂMICA</b>  | <b>137</b> |
| <b>CORRELAÇÃO COM A NEUROBIOLOGIA. IMAGENS ESTABILIZADAS NA RETINA.</b>  | <b>143</b> |
| <b>EXPLICAÇÃO DETALHADA DE CADA MÓDULO DO PROGRAMA:</b>  | <b>149</b> |
| ▪ Programa: PCAsinc_oscl   | 149        |
| ▪ Programa: Coge_el_patron   | 155        |
| ▪ Programa: Prob_max_ret_f   | 158        |
| ▪ Programa: Prob_rele_f  | 160        |
| ▪ Programa: Stim_retic_f   | 161        |

|   |            |
|---|------------|
| ▪ Programa: mod_pesos_f                   | 162        |
| ▪ Programa: calculate_weights             | 163        |
| ▪ Programa: GraficoPCAsincosc             | 164        |
| ▪ Programa: grafico_red.                  | 166        |
| ▪ Programa: max2D                         | 168        |
| ▪ Programa: sigmoid1                      | 169        |
| ▪ Programa: linear                        | 170        |
| ▪ Programa: Muestra_pesos                 | 171        |
| ▪ Programa: Muestra_pesos_tr              | 172        |
| ▪ Programa: stabilized_images             | 173        |
| ▪ Programa: GraficoPCAsincosc_s_i         | 176        |
| <b>Aplicações e perspectivas futuras:</b> | <b>178</b> |
| <b>Anexo 1</b>                            | <b>180</b> |
| <b>Bibliografia.</b>                      | <b>183</b> |

## **INTRODUÇÃO:**

Esta tese de doutorado consta de três partes. A Parte I consiste numa série de novos desenvolvimentos matemáticos realizados inteiramente pelo doutorando que permitem embasar as duas Partes seguintes. Esta primeira parte matemática é uma nova visão da estatística sob o ponto de vista da álgebra linear. A partir deste ponto de vista, qualquer evento estatístico é considerado um vetor. A probabilidade de um evento é uma medida, determinada pelo valor escalar da projeção deste evento-vetor sobre outro evento-vetor que será descrito mais tarde. Nesta primeira parte, os teoremas e conceitos mais importantes da estatística serão demonstrados sob uma perspectiva geométrica. Por exemplo serão demonstrados o teorema de Bayes, o teorema da probabilidade total, a regra de decisão de Bayes, a probabilidade da interseção de eventos, o conceito de independência de eventos, etc. Esta nova perspectiva facilita enormemente os cálculos estatísticos que serão baseados em transformações lineares, projeções, obtenção de eixos ortogonais a partir de um conjunto de vetores não ortogonais etc, como será demonstrado num exemplo de cálculo de probabilidades em urnas. Esta perspectiva algébrica facilita também o desenvolvimento de novos teoremas estatísticos como uma nova versão estatística do teorema de Gram-Schmidt para extração de eixos ortogonais a partir de um conjunto de eixos não ortogonais. No teorema de Gram-Schmidt a seqüência dos vetores não ortogonais é muito importante para a obtenção de eixos. Contudo temos interesse em achar um conjunto de eixos que independa da seqüência dos vetores. Para conseguir este objetivo

derivaremos uma nova versão do teorema de Gram-Schmidt que não depende da seqüência de vetores de entrada. Posteriormente e, seguindo a mesma linha de pensamento, encontraremos um conjunto de eixos que além de serem ortogonais representam os chamados componentes principais do conjunto inicial de vetores (ou padrões).

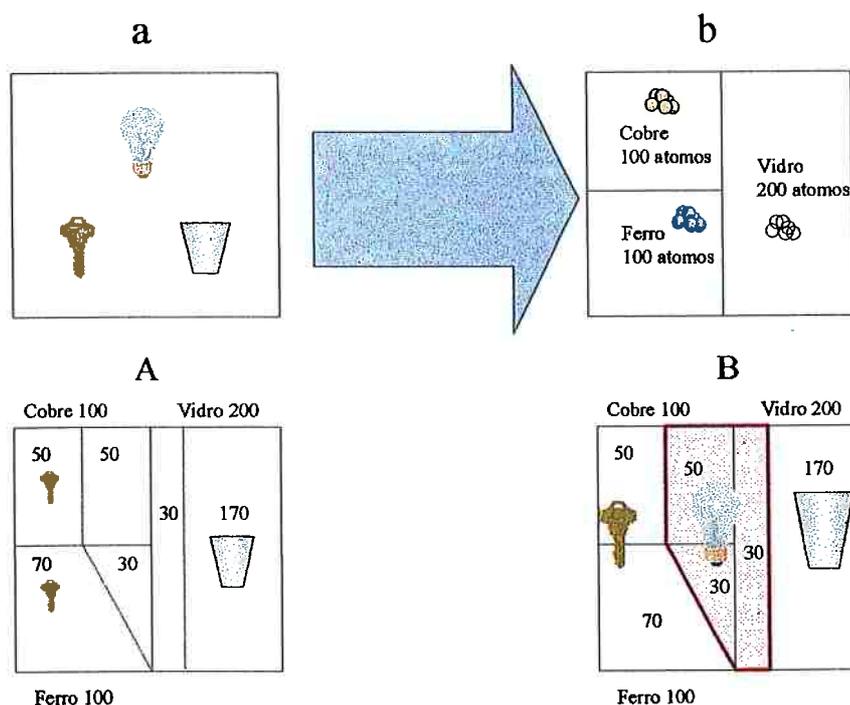
As Partes II e III da tese consistem em aplicar estas idéias a duas áreas da inteligência artificial: a lógica *fuzzy* e as redes neurais. Fica em destaque com este fato a importância do embasamento estatístico destas duas disciplinas. A nova ferramenta matemática desenvolvida permite esclarecer alguns pontos que pareciam arbitrários ou exclusivamente de caráter empírico na lógica *fuzzy*. Por exemplo, explicar-se-á a razão das funções de pertinência serem triangulares ou trapezoidais derivando regras analíticas para o cálculo da forma destas funções de pertinência. Também serão analisados os chamados processos de *fuzzificação* e *defuzzificação* como transformações lineares dentro dos chamados espaços euclidiano-probabilísticos apresentados na Parte I. Por outra parte, esta ferramenta matemática pode representar uma extraordinária ajuda para o desenho de redes neurais e para a análise de redes existentes incluindo as próprias redes biológicas. Por este último motivo, esta matemática será aplicada ao desenvolvimento de uma rede neural baseada na arquitetura do tálamo, estrutura localizada no centro do cérebro que parece ‘filtrar’ e comprimir[16, 19, 39] as informações sensoriais antes destas serem armazenadas no córtex cerebral. Propõe-se que esta rede faça uma extração dos componentes principais do conjunto de padrões sensoriais. Com isto se consegue uma importante compressão de dados e uma otimização da representação dos mesmos num espaço de dimensão menor que o espaço de

entrada da informação sensorial. Este modelo, pelas suas próprias características, pode ser aplicado na área da engenharia a tarefas de extração de atributos para reconhecimento de padrões, compressão e criptografia de dados, etc.

# 1. PARTE 1. NOVOS EMBASAMENTOS DA ESTATÍSTICA SOB O PONTO DE VISTA DA ÁLGEBRA LINEAR.

*Eventos estatísticos considerados como vetores num espaço Euclídiano.*

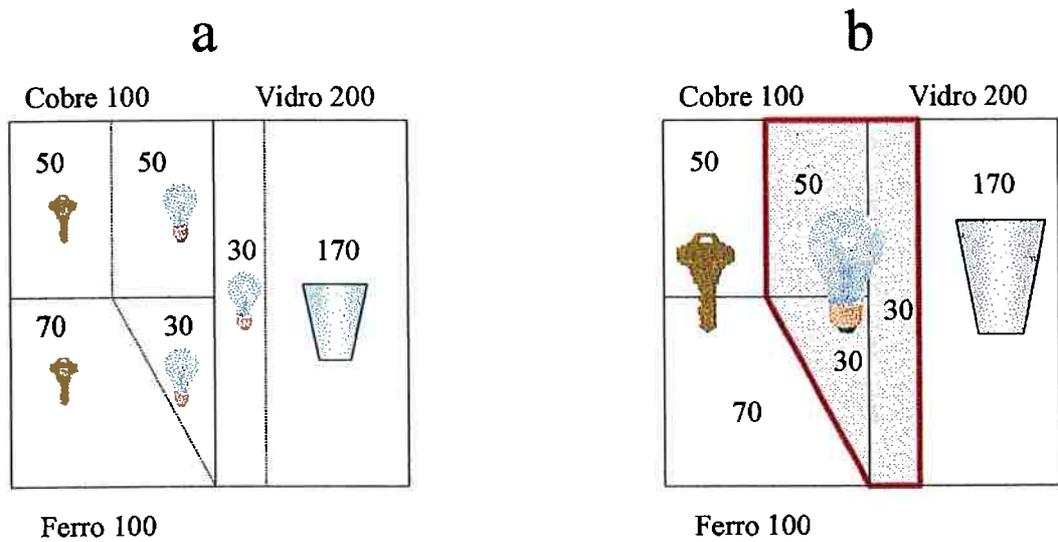
Imaginemos um universo  $U$  somente com três objetos: um copo, uma chave e uma lâmpada (Figura 1.a). Se este universo for reduzido a "átomos", reais ou fictícios, isto nos conduzirá a situação da Figura 1.b na qual os "átomos" são agrupados em categorias. Portanto este universo tem 100 "átomos" de cobre, 100 "átomos" de ferro e 200 "átomos" de vidro.



**Figura 1** Neste exemplo, um simples universo é reduzido a "átomos". Estes "átomos" são

agrupados de acordo com seus tipos, contando-se o número de “átomos” em cada categoria.

Uma classificação adicional pode ser feita, distribuindo-se os “átomos” em distintas categorias como na Figura 2.a. Por exemplo, a quantidade total de 100 “átomos” de cobre é distribuída da seguinte maneira: 50 “átomos” de cobre pertencentes a chave e os outros 50 “átomos” de cobre pertencentes a lâmpada.



**Figura 2** Cada categoria de “átomos”, é classificada de acordo com o objeto a que eles pertencem. Desta forma é possível descrever cada objeto segundo as proporções das diversas categorias no objeto.

Esta classificação permite descrever cada um dos três objetos levando em consideração somente a composição atômica sem considerar outras características como forma, tamanho, etc. Desta forma obtém-se as seguintes definições para cada objeto:

$$\begin{aligned}
Chave &= \frac{50}{100}Cobre + \frac{70}{100}Ferro + \frac{0}{200}Vidro \\
L\u00e2mpada &= \frac{50}{100}Cobre + \frac{30}{100}Ferro + \frac{30}{200}Vidro \\
Copo &= \frac{0}{100}Cobre + \frac{0}{100}Ferro + \frac{170}{200}Vidro
\end{aligned}
\tag{1}$$

na qual cada categoria de “\u00e1tomos” \u00e9 multiplicada por um fator que representa as diferentes propor\u00e7\u00f5es na qual este objeto comparte os diversos tipos de “\u00e1tomos” neste universo. Neste sentido, usando a terminologia da l\u00f3gica *fuzzy* [38], cada fator indica o grau de pertin\u00eancia dos eventos chave, l\u00e2mpada e copo de vidro ao conjunto cobre, ferro e vidro. Usando a mesma terminologia, cada uma das tr\u00eas descri\u00e7\u00f5es acima \u00e9 um conjunto *fuzzy* dentro do seu “universo de discurso”.

Usemos, por exemplo, a defini\u00e7\u00e3o de l\u00e2mpada para obter uma express\u00e3o geral a fim de descrever qualquer objeto, vide Figura 2.b. Com este prop\u00f3sito o operador  $n(\ )$ , n\u00famero de vezes, ser\u00e1 usado com o prop\u00f3sito de computar quantas vezes o conceito de l\u00e2mpada intersecta cada categoria de \u00e1tomo.

$$L\u00e2mpada = \frac{n(L\u00e2mpada \cap Cobre)}{n(Cobre)}Cobre + \frac{n(L\u00e2mpada \cap Ferro)}{n(Ferro)}Ferro + \frac{n(L\u00e2mpada \cap Vidro)}{n(Vidro)}Vidro
\tag{2}$$

Note que cada fator, por exemplo,  $n(L\u00e2mpada \cap Cobre) / n(Cobre)$  \u00e9 equivalente a probabilidade condicional  $P(L\u00e2mpada/Cobre)$ .

Apesar de que o conceito probabilidade ser\u00e1 redefinido posteriormente estamos mencionando aqui probabilidade condicional para permitir que o leitor se acostume com a id\u00e9ia da rela\u00e7\u00e3o entre vetores e probabilidade. Para

homogeneizar numerador e denominador na equação 2, o denominador é transformado da seguinte forma:

$$L\grave{a}mpada = \frac{n(L\grave{a}mpada \cap Cobre)}{n(Cobre \cap Cobre)} Copper + \frac{n(L\grave{a}mpada \cap Ferro)}{n(Ferro \cap Ferro)} Ferro + \frac{n(L\grave{a}mpada \cap Vidro)}{n(Vidro \cap Vidro)} Vidro \quad (3)$$

Comparemos esta express\~ao com a express\~ao gen\~erica de um vetor  $\vec{B}$  em fun\~cao de tr\~es componentes  $\vec{Y}_i$ , como normalmente \~e descrito em qualquer livro de \~algebra linear.

$$\vec{B} = \frac{(\vec{B}, \vec{Y}_1)}{(\vec{Y}_1, \vec{Y}_1)} \vec{Y}_1 + \frac{(\vec{B}, \vec{Y}_2)}{(\vec{Y}_2, \vec{Y}_2)} \vec{Y}_2 + \frac{(\vec{B}, \vec{Y}_3)}{(\vec{Y}_3, \vec{Y}_3)} \vec{Y}_3 \quad (4)$$

Aqui  $(\vec{B}, \vec{Y}_i)$  significa produto interior ou escalar do vetor  $\vec{B}$  e do vetor  $\vec{Y}_i$ . Pode-se ver que a equa\~ao 2 \~e igual a equa\~ao acima, considerando-se  $n(\vec{B} \cap \vec{Y}_i)$  como um produto interior. Produtos interiores s\~ao freq\~uentemente definidos como  $\|\vec{B}\| \|\vec{Y}_i\| \cos(\alpha)$ , sendo isto apenas um caso particular da defini\~ao gen\~erica de produto escalar. A defini\~ao de um produto escalar parte do princ\~ipio de que qualquer elemento B ou Y pertence a um espa\~co linear e que a opera\~ao definida entre dois vetores segue as propriedades comutativa, distributiva, associativa e de positividade. Como os eventos considerados no exemplo acima cumprem com os dez axiomas que definem um espa\~co linear, e a opera\~ao entre eles est\~a de acordo com quatro axiomas, os quais definem produto interior, a opera\~ao  $n(\vec{B} \cap \vec{Y}_i)$  pode ser compreendida como um produto interior. Todo espa\~co linear de vetores no qual \~e definido um produto interior \~e chamado espa\~co Euclidiano. O leitor interessado pode encontrar as propriedades e axiomas dos espa\~cos lineares e produto escalar o interior no Anexo 1.

As propriedades acima caracterizam os eventos do nosso universo U, como vetores em um espaço Euclidiano. Por esta razão o símbolo de vetor é colocado sobre cada componente da equação 2.

$$\vec{Lâmpada} = \frac{n(\vec{Lâmpada} \cap \vec{Cobre})}{n(\vec{Cobre})} \vec{Cobre} + \frac{n(\vec{Lâmpada} \cap \vec{Ferro})}{n(\vec{Ferro})} \vec{Ferro} + \frac{n(\vec{Lâmpada} \cap \vec{Vidro})}{n(\vec{Vidro})} \vec{Vidro} \quad (5)$$

No caso de um evento genérico  $\vec{B}$ , esta expressão é escrita desta forma:

$$\vec{B} = \sum_{i=1}^n \frac{n(\vec{B} \cap \vec{Y}_i)}{n(\vec{Y}_i \cap \vec{Y}_i)} \cdot \vec{Y}_i \quad (6)$$

A equação anterior pode ser escrita também em função das probabilidades condicionais:

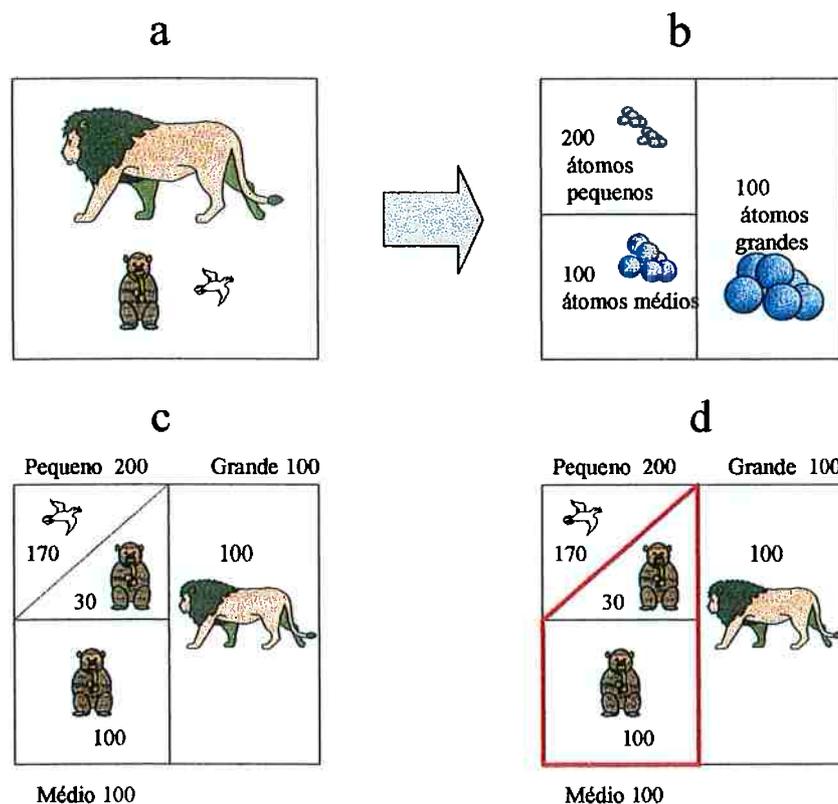
$$\vec{B} = \sum_{i=1}^n P(\vec{B} / \vec{Y}_i) \cdot \vec{Y}_i \quad (7)$$

Cada um dos  $\vec{Y}_i$  representa um eixo ortogonal de uma base de eixos coordenados em um espaço linear. Eles são ortogonais um ao outro porque seu produto interior é zero.

$$n(\vec{Y}_i \cap \vec{Y}_j) = 0, \forall i \neq j \quad (8)$$

Usando a terminologia da lógica *fuzzy* cada um destes eixos representa um conjunto “crisp”.

É também possível aplicar a filosofia acima de representação de eventos, dividindo-se estes eventos em seus “átomos” constituintes, mas usando “átomos” fictícios que representariam características macroscópicas ou subjetivas como tamanho, cor, beleza, bondade etc. Desta forma consideremos o próximo universo de discurso, do qual fazem parte três tipos de animais: um leão, um macaco e um pombo (Figura 3).



**Figura 3** Os objetos podem ser identificados não somente através da característica físicas dos “átomos” que os constituem, mas também por características subjetivas relacionadas aos “átomos”, como por exemplo tamanho. Tendo-se “átomos” de diferentes tamanhos (pequeno, médio e grande), podemos descrever todos os animais deste simples universo acima representado, através da combinação, em diferentes proporções, de cada tipo de átomo.

Suponhamos que o tamanho é a única característica pela qual estes animais podem ser descritos. Como no exemplo anterior, estes animais são reduzidos à “átomos”, mas neste caso “átomos” com a característica “tamanho” (Figura 3.B). Neste caso o universo está composto de 100 “átomos” grandes, 100 “átomos” médios e 100 “átomos” pequenos. Da mesma forma que anteriormente, os eventos são tratados como vetores:

$$\begin{aligned} \vec{Macaco} &= \frac{n(\vec{Macaco} \cap \vec{Pequeno})}{n(\vec{Pequeno})} \vec{Pequeno} + \frac{n(\vec{Macaco} \cap \vec{Médio})}{n(\vec{Médio})} \vec{Médio} \\ &+ \frac{n(\vec{Macaco} \cap \vec{Grande})}{n(\vec{Grande})} \vec{Grande} = P(\vec{Macaco} / \vec{Pequeno}) \vec{Pequeno} + \\ &P(\vec{Macaco} / \vec{Médio}) \vec{Médio} + P(\vec{Macaco} / \vec{Grande}) \vec{Grande} \end{aligned} \quad (9)$$

A definição de evento pode ser ainda mais geral considerando-se os átomos como a origem ou a causa do próprio evento. Considere-se a situação de um automóvel A, manufaturado por três fabricas diferentes.

$$\vec{A} = \frac{n(\vec{A} \cap \vec{F}_1)}{n(\vec{F}_1)} \cdot \vec{F}_1 + \frac{n(\vec{A} \cap \vec{F}_2)}{n(\vec{F}_2)} \cdot \vec{F}_2 + \frac{n(\vec{A} \cap \vec{F}_3)}{n(\vec{F}_3)} \cdot \vec{F}_3 \quad (10)$$

Seja qual for a natureza de cada um dos átomos dos quais esta composto cada “evento”, é muito importante salientar que os eventos representados (chave, lâmpada, copo, leão, carro) não devem ser considerados estritamente como objetos individuais. Por exemplo o vetor copo representa não um, mas a todos os copos construídos no presente, passado e futuro. Assim combinando os 200 átomos de vidro e possível obter um mínimo de  $\binom{200}{170}$  concretizações diferentes do conceito genérico, (tipo ou “template”) que chamaremos “copo”. Contudo um copo poderia ter mais ou menos de 170 átomos de vidro sem perder a caraterística de copo. O número 170 e simplesmente uma média dentre todas as concretizações possíveis do conceito copo.

*Por isto o vetor “copo” tem ao mesmo tempo duas acepções :*

*A/ Acepção populacional: O vetor copo representando a todos e cada um dos copos que já existiram, existem e existirão.*

*B/Acepção ideal. O vetor copo na sua apresentação matemática, como combinação linear de um conjunto de átomos ou vetores ortogonais. Nesta acepção os componentes são valores médios após promediar as diversas concretizações do vetor copo ao longo do tempo.*

A questão de quando um copo deixa de ser copo para ser lâmpada, por exemplo, é uma questão de distância ou, mais rigorosamente, de ângulo. Quando o ângulo entre um vetor desconhecido e o vetor “copo”, por exemplo, é o menor, então consideraremos o vetor desconhecido sendo do tipo “copo” e não lâmpada ou chave. Uma concretização específica do vetor “copo” poderia ser entendida como a soma do vetor genérico (o vetor médio) “copo” mais um ou vários vetores ortogonais a ele que representaria o desvio do copo concreto em relação ao copo genérico ou médio. Deste modo podemos perceber a grande utilidade que poderiam ter os chamados componentes principais (direções de maior variabilidade num conjunto de padrões) para descrever um vetor concreto a partir de um outro genérico. Estes componentes principais serão analisadas nas partes 2 e 4 da tese. Nas seguintes secções, nos aprofundaremos nos conceitos de projecção de um vetor sobre outro e veremos como o próprio conceito de probabilidade é uma projecção (A projecção de um vetor unitário sobre um outro constitui uma forma de avaliar a distância angular entre os dois)

### ***Probabilidade como uma projeção.***

Se um vetor unitário  $\vec{y}_i$  (em minúscula) é definido como:

$$\vec{y}_i = \frac{\vec{Y}_i}{n(\vec{Y}_i)} \quad (11)$$

a equação 6 é escrita da seguinte forma:

$$\begin{aligned} \vec{B} &= \sum_{i=1}^n \frac{n(\vec{B} \cap \vec{Y}_i)}{n(\vec{Y}_i)} \cdot \vec{Y}_i = \sum_{i=1}^n n \left( \vec{B} \cap \frac{\vec{Y}_i}{n(\vec{Y}_i)} \right) \cdot \vec{Y}_i = \\ &= \sum_{i=1}^n n(\vec{B} \cap \vec{y}_i) \cdot \vec{Y}_i = \sum_{i=1}^n \vec{B}_{\vec{Y}_i} \end{aligned} \quad (12)$$

portanto:

$$\vec{B}_{\vec{Y}_i} = n(\vec{B} \cap \vec{y}_i) \cdot \vec{Y}_i \quad (13)$$

Desta forma, o fator que multiplica cada categoria de “átomos”, é definido como produto interior de um determinado evento  $\vec{B}$  com o vetor unitário  $\vec{y}_i$ . De acordo com a álgebra linear,  $\vec{B}_{\vec{Y}_i}$  representa a projeção do vetor  $\vec{B}$  sobre cada um dos eixos  $\vec{Y}_i$ .

A equação acima pode ser escrita da seguinte forma:

$$\vec{B}_{\vec{Y}_i} = n(\vec{B} \cap \vec{y}_i) \vec{Y}_i = P(\vec{B} / \vec{Y}_i) \vec{Y}_i \quad (14)$$

No primeiro exemplo, um vetor unitário de cobre, pode ser definido da seguinte forma (onde as letras minúsculas representam vetores unitários):

$$cobre = \frac{Cobre}{|n(Cobre)|} = \frac{Cobre}{100} \quad (15)$$

e a projeção do evento lâmpada sobre o evento cobre é descrito à seguir:

$$\begin{aligned}
Lâmpada_{Cobre} &= n(Lâmpada \cap cobre) \cdot Cobre = \\
&= n\left(Lâmpada \cap \frac{Cobre}{n(Cobre)}\right) \cdot Cobre = \\
&= n\left(Lâmpada \cap \frac{Cobre}{100}\right) \cdot Cobre = \\
&= \frac{1}{100} n(Lâmpada \cap Cobre) \cdot Cobre = \\
&= \frac{50}{100} Cobre = 0.5 \cdot Cobre
\end{aligned} \tag{16}$$

Desta forma o evento lâmpada pode ser expresso em termos de suas projeções sobre os seus “átomos” constituintes, desta forma:

$$\overrightarrow{Lâmpada} = \overrightarrow{Lâmpada}_{Cobre} + \overrightarrow{Lâmpada}_{Ferro} + \overrightarrow{Lâmpada}_{Vidro} \tag{17}$$

Foi mostrado na equação 13 que uma determinada medida  $P(\vec{B}/\vec{Y}_i)$  pode ser considerada como a projeção do vetor  $\vec{B}$  sobre um dos seus eventos atômicos constituintes. Seguindo este raciocínio estudaremos a seguir uma hipótese que considera a probabilidade como uma projeção sobre um vetor específico. Este vetor específico  $\vec{u}$  está na direção do universo  $\vec{U}$ , o qual é a soma de todos os eixos  $\vec{Y}_i$ . Portanto o vetor  $\vec{u}$  é obtido da seguinte maneira:

$$\vec{u} = \frac{\vec{U}}{n(\vec{U})} = \frac{\sum_{i=1}^n \vec{Y}_i}{\sum_{i=1}^n n(\vec{Y}_i)} \tag{18}$$

onde  $n(\vec{U})$  é a cardinalidade deste espaço amostral. Desta forma, o vetor  $\vec{u}$  é o universo normalizado ou, em outras palavras, é o vetor espaço amostral normalizado.

Agora a projeção do vetor  $\vec{B}$  sobre o vetor  $\vec{u}$  é calculado a fim de se demonstrar que a probabilidade de  $\vec{B}$  é o valor desta projeção.

$$n(\bar{B} \cap \bar{u}) = n\left(\bar{B} \cap \frac{\bar{U}}{n(\bar{U})}\right) = \frac{n(\bar{B} \cap \bar{U})}{n(\bar{U})} = \frac{n(\bar{B})}{n(\bar{U})} = P(\bar{B}) \quad (19)$$

A expressão do teorema da probabilidade total  $P(\bar{B}) = \sum_{i=1}^n P(\bar{B}/\bar{Y}_i)P(\bar{Y}_i)$ , pode ser derivada calculando-se  $n(\bar{B} \cap \bar{u})$ .

$$\begin{aligned} P(\bar{B}) = n(\bar{B} \cap \bar{u}) &= n\left(\left[\sum_{i=1}^n P(\bar{B}/\bar{Y}_i) \cdot \bar{Y}_i\right] \cap \frac{\sum_{i=1}^n \bar{Y}_i}{\sum_{i=1}^n n(\bar{Y}_i)}\right) = a \\ &= n\left(\sum_{i=1}^n P(\bar{B}/\bar{Y}_i) \cdot \frac{\bar{Y}_i \cap \bar{Y}_i}{\sum_{i=1}^n n(\bar{Y}_i)}\right) = \\ &= \sum_{i=1}^n P(\bar{B}/\bar{Y}_i) \cdot \frac{n(\bar{Y}_i \cap \bar{Y}_i)}{\sum_{i=1}^n n(\bar{Y}_i)} = \sum_{i=1}^n P(\bar{B}/\bar{Y}_i) \cdot \frac{n(\bar{Y}_i)}{n(\bar{U})} = b \\ &= \sum_{i=1}^n P(\bar{B}/\bar{Y}_i) \cdot P(\bar{Y}_i) \end{aligned} \quad (20)$$

Temos abaixo explicações para as etapas *a* e *b* da equação acima:

(a) Partindo do princípio de que  $\forall i \neq j \quad n(\bar{Y}_i \cap \bar{Y}_j) = 0$

(b) O segundo fator calcula a probabilidade de  $\bar{Y}_i$ .

Observe que dentro da definição da probabilidade total de um evento B

(equação 19) partimos do princípio de que  $P(\bar{Y}_i) = n(\bar{Y}_i) / \sum_{i=1}^n n(\bar{Y}_i)$ . Neste caso a

definição de probabilidade de  $\bar{Y}_i$  é também a projeção do vetor  $\bar{Y}_i$  sobre  $\bar{u}$ , e é calculada como  $n(\bar{Y}_i \cap \bar{u})$ :

$$P(\vec{Y}_i) = n(\vec{Y}_i \cap \vec{u}) = \frac{n(\vec{Y}_i \cap \vec{Y}_i)}{n(\vec{Y}_1) + n(\vec{Y}_2) + \dots + n(\vec{Y}_n)} = \frac{n(\vec{Y}_i)}{n(\vec{U})} \quad (21)$$

Nos parágrafos anteriores, a expressão de probabilidade condicional  $P(B/Y_i)$  foi utilizada. Contudo não foi explicado se esta probabilidade condicional também é definida através de uma projeção. A fim de calcular a probabilidade condicional,  $\vec{B}$  é projetada sobre  $\vec{Y}_i$  usando-se  $n(\vec{B} \cap \vec{y}_i)$ , como se  $\vec{Y}_i$  fosse o  $\vec{u}$ , ou seja como se  $\vec{Y}_i$  fosse o espaço amostral normalizado de um universo unidimensional. Portanto a probabilidade condicional é também uma projeção mas, neste caso, sobre um vetor unitário que pertence a um universo unidimensional.

O teorema a seguir, resume as explicações anteriores:

*Teorema:*

*A probabilidade de um evento  $\vec{B}$  é igual a projeção de  $\vec{B}$ ,  $n(\vec{B} \cap \vec{u})$ , sobre o evento unitário  $\vec{u}$  em um espaço  $n$  dimensional  $U^n$ , onde  $\vec{u}$  é definido como:*

$$\vec{u} = \frac{\vec{U}}{n(\vec{U})} = \frac{\sum_{i=1}^n \vec{Y}_i}{n(\vec{U})} = \frac{\sum_{i=1}^n \vec{Y}_i}{n(\sum_{i=1}^n \vec{Y}_i)} = \frac{\sum_{i=1}^n \vec{Y}_i}{\sum_{i=1}^n n(\vec{Y}_i)} \quad (22)$$

sendo a operação  $n(x \cap y)$  um produto escalar definido no espaço euclidiano de eventos. Um espaço Euclidiano onde tem-se definido este tipo de projeção será chamado a partir deste momento espaço Euclidiano-probabilístico.

Como foi visto acima, qualquer tipo de vetor dentro de um espaço Euclidiano-probabilístico, ou em outras palavras, qualquer tipo de conjunto *fuzzy* dentro de seu universo de discurso, pode ser medido usando-se suas projeções sobre um vetor unitário específico. Desta forma, o universo de

eventos qualitativos, com o qual a lógica *fuzzy* costuma tratar, pode ser medido através de teoremas estatísticos derivados da álgebra euclidiana, como sera explicado na Parte II.

**Probabilidade da interseção de eventos.**

Nesta seção a probabilidade da interseção de dois eventos  $\bar{X}$  e  $\bar{A}$ ,  $P(\bar{X} \cap \bar{A})$  será calculada em termos de seus componentes  $\bar{Y}_i$ . Para começar, expressaremos a interseção de dois padrões  $\bar{X} \cap \bar{A}$  em termos de  $\bar{Y}_i$ , expressando-se cada padrão como na equação 6.

$$\sum_{i=1}^{i=n} P(\bar{X} / \bar{Y}_i) \bar{Y}_i \cap \sum_{i=1}^{i=n} P(\bar{A} / \bar{Y}_i) \bar{Y}_i \quad (23)$$

Anulando-se a interseção entre padrões ortogonais  $\forall i \neq j, \bar{Y}_i \cap \bar{Y}_j = 0$ , obtém-se:

$$\begin{aligned} \bar{X} \cap \bar{A} &= \sum_{i=1}^{i=n} (P(\bar{X} / \bar{Y}_i) \bar{Y}_i \cap P(\bar{A} / \bar{Y}_i) \bar{Y}_i) = \\ &= \sum_{i=1}^n (\bar{X}_{\bar{Y}_i} \cap \bar{A}_{\bar{Y}_i}) \end{aligned} \quad (24)$$

ou, pela propriedade associativa do produto escalar:

$$\begin{aligned} \bar{X} \cap \bar{A} &= \sum_{i=1}^{i=n} (P(\bar{X} / \bar{Y}_i) \bar{Y}_i \cap P(\bar{A} / \bar{Y}_i) \bar{Y}_i) = \\ &= \sum_{i=1}^n [P(\bar{X} / \bar{Y}_i) P(\bar{A} / \bar{Y}_i)] \bar{Y}_i \end{aligned} \quad (25)$$

A probabilidade da interseção é calculada mediante o produto interno da interseção com o vetor  $\bar{u}$ .

$$\begin{aligned} P(\bar{X} \cap \bar{A}) &= n(\bar{X} \cap \bar{A}) \cap \bar{u} = n \left( \left[ \sum_{i=1}^n (\bar{X}_{\bar{Y}_i} \cap \bar{A}_{\bar{Y}_i}) \right] \cap \bar{u} \right) = \\ &= n(\bar{X}_{\bar{Y}_1} \cap \bar{A}_{\bar{Y}_1}) \cap \bar{u} + n(\bar{X}_{\bar{Y}_2} \cap \bar{A}_{\bar{Y}_2}) \cap \bar{u} + \dots + n(\bar{X}_{\bar{Y}_n} \cap \bar{A}_{\bar{Y}_n}) \cap \bar{u} = \sum_{i=1}^n P(\bar{X}_{\bar{Y}_i} \cap \bar{A}_{\bar{Y}_i}) \end{aligned} \quad (26)$$

Que significa que as probabilidades da interseção podem ser calculadas

sobre cada eixo e em seguida somadas.

Consideremos  $\vec{X} \cap \vec{A}$  como se fosse um só vetor e utilizemos a equação 20:

$$P(\vec{X} \cap \vec{A}) = \sum_{i=1}^n P((\vec{X} \cap \vec{A}) / \vec{Y}_i) P(\vec{Y}_i) \quad (27)$$

E, segundo a equação 24 :

$$\begin{aligned} P(\vec{X} \cap \vec{A}) &= \sum_{i=1}^n (P(\vec{X} / \vec{Y}_i) \vec{Y}_i \cap P(\vec{A} / \vec{Y}_i) \vec{Y}_i) = \\ &= \sum_{i=1}^n [P(\vec{X} / \vec{Y}_i) P(\vec{A} / \vec{Y}_i)] P(\vec{Y}_i) \end{aligned} \quad (28)$$

Utilizando as duas equações anteriores:

$$P((\vec{X} \cap \vec{A}) / \vec{Y}_i) = P(\vec{X} / \vec{Y}_i) P(\vec{A} / \vec{Y}_i) \quad (29)$$

É também possível generalizar o resultado da equação 25 calculando a expressão do padrão  $\vec{\cap} B$ , que é a interseção de um número “m” de outros padrões, para depois encontrar  $P(\vec{\cap} B)$ :

$$\vec{\cap} B = \vec{\cap}_{j=1}^m \vec{B}^j = \sum_{i=1}^n \left[ \vec{\cap}_{j=1}^m \vec{B}_{\vec{Y}_i}^j \right] \quad (30)$$

$$\begin{aligned} P(\vec{\cap} B) &= P(\vec{\cap}_{j=1}^m \vec{B}^j) = \sum_{i=1}^n P\left(\vec{\cap}_{j=1}^m \vec{B}^j / \vec{Y}_i\right) P(\vec{Y}_i) = \\ &= \sum_{i=1}^n \left( \prod_{j=1}^m P(\vec{B}^j / \vec{Y}_i) \right) P(\vec{Y}_i) \end{aligned} \quad (31)$$

de maneira que a probabilidade condicionada de um padrão composto, em termos das probabilidades condicionadas dos seus componentes pode ser escrito como:

$$P\left(\vec{\cap}_{j=1}^m \vec{B}^j / \vec{Y}_i\right) = \prod_{j=1}^m P(\vec{B}^j / \vec{Y}_i) \quad (32)$$

Algumas considerações, que poderiam parecer óbvias, em relação à interseção de eventos, devem ser realizadas neste momento. Além disto podem ajudar a esclarecer possíveis dúvidas que poderiam aparecer no resto desta tese.

A interseção consigo mesmo de eventos incompatíveis ou exclusivos  $\vec{Y}_i$  é o próprio evento:

$$\vec{Y}_i \cap \vec{Y}_i = \vec{Y}_i \quad (33)$$

pois:

$$\vec{Y}_i \cap \vec{Y}_i = P(\vec{Y}_i / \vec{Y}_i) \vec{Y}_i \cap P(\vec{Y}_i / \vec{Y}_i) \vec{Y}_i = P(\vec{Y}_i / \vec{Y}_i) \cdot P(\vec{Y}_i / \vec{Y}_i) \vec{Y}_i = 1 \cdot 1 \cdot \vec{Y}_i$$

Este tipo de igualdade também acontece com o vetor  $\vec{U}$  que representa o universo ou espaço amostral.

$$\vec{U} \cap \vec{U} = \left( \sum_{i=1}^n P(\vec{U} / \vec{Y}_i) \vec{Y}_i \right) \cap \left( \sum_{i=1}^n P(\vec{U} / \vec{Y}_i) \vec{Y}_i \right) = \left( \sum_{i=1}^n P(\vec{Y}_i / \vec{Y}_i) \vec{Y}_i \right) \cap \left( \sum_{i=1}^n P(\vec{Y}_i / \vec{Y}_i) \vec{Y}_i \right) = \left( \sum_{i=1}^n 1 \cdot \vec{Y}_i \right) \cap \left( \sum_{i=1}^n 1 \cdot \vec{Y}_i \right) = \sum_{i=1}^n \vec{Y}_i = \vec{U} \quad (34)$$

Mas em geral a interseção de um evento consigo mesmo é diferente do próprio evento:

$$\vec{A} \cap \vec{A} \neq \vec{A} \quad (35)$$

Calculemos esta interseção:

$$\begin{aligned} \vec{A} \cap \vec{A} &= \sum_{i=1}^{i=n} (P(\vec{A} / \vec{Y}_i) \vec{Y}_i \cap P(\vec{A} / \vec{Y}_i) \vec{Y}_i) = \\ &= \sum_{i=1}^n [P(\vec{A} / \vec{Y}_i) P(\vec{A} / \vec{Y}_i)] \vec{Y}_i \end{aligned} \quad (36)$$

As equações anteriores podem ser melhor compreendidas se voltarmos ao exemplo dos três objetos: copo, chave e lâmpada. Seja por exemplo o copo constituído por 170 “átomos”, dos duzentos “átomos de vidro” deste pequeno universo. Com os duzentos “átomos” poderiam ser formados distintos copos

de 170 “átomos”, cada um distinto do outro. Portanto nunca poderíamos garantir que dos copos formados pelos “átomos” deste universo são exatamente iguais ao menos que cada copo estivesse formado pelos duzentos “átomos” de vidro que compõem o universo. Por esse motivo a interseção de dois eventos ‘aparentemente’ iguais não é, em geral igual ao próprio evento. Demonstrar-se a, em seguida, que a interseção de dois eventos (tipos) complementares não é em geral igual a zero:  $\vec{A} \cap \vec{A}^c \neq 0$ . Isto parece incompatível com os princípios da probabilidade Bayesiana. Contudo esta desigualdade constitui o fundamento da lógica *fuzzy*.

Tendo em vista que  $\vec{U} = \vec{A} + \vec{A}^c$ , e que, segundo a última demonstração  $\vec{A} \cap \vec{A}$  se diferencia de  $\vec{A}$  numa quantidade  $\zeta$ :  $\vec{A} \cap \vec{A} = \vec{A} + \zeta$  podemos calcular:

$$\vec{A} \cap \vec{A}^c = \vec{A} \cap (\vec{U} - \vec{A}) = \vec{A} \cap \vec{U} - \vec{A} \cap \vec{A} = \vec{A} - (\vec{A} + \zeta) = -\zeta \neq 0 \quad (37)$$

Unicamente quando o evento e o seu complementar se projetam sobre grupos de eixos diferentes, a interseção destes eventos é nula. Com esta nova perspectiva os eventos complementares (definidos como  $\vec{A}^c = \vec{U} - \vec{A}$ ) poderiam-se sub-dividir em dois tipos: os eventos ortogono-complementares, cuja interseção é zero, e os eventos complementares não ortogonais cuja interseção é diferente de zero.

### *Um enfoque algébrico para a regra de decisão de Bayes*

Nesta seção serão demonstrados o teorema e a regra de decisão de Bayes[2, 4, 6, 36], utilizando-se as novas idéias algébricas apresentadas. Enquanto o teorema de Bayes é um dos teoremas mais importantes da estatística, a regra de decisão de Bayes vem sendo utilizada na área de reconhecimento de padrões para classificar um determinado padrão  $X$  como pertencente a um entre vários grupos  $A, B, C$  etc. A regra de decisão de Bayes classifica a  $X$  como pertencente ao grupo que maximiza  $P(X \cap Grupo)$ . Nos seguintes parágrafos será demonstrado que a probabilidade da projeção de cada grupo sobre  $X$  é equivalente a equação  $P(X \cap Grupo)$ . Com isto conseguiremos dar um enfoque mais geométrico à regra de decisão de Bayes. Posteriormente será demonstrado o próprio teorema de Bayes. Para conseguir estes objetivos começaremos estudando a projeção de um vetor sobre outro.

A projeção de um determinado vetor  $\vec{B}$  sobre um outro  $\vec{A}$ ,  $\vec{B}_{\vec{A}}$ , expressa-se na álgebra linear com a seguinte equação:

$$\vec{B}_{\vec{A}} = \frac{(\vec{B}, \vec{A})}{(\vec{A}, \vec{A})} \cdot \vec{A} \quad (38)$$

Substituindo-se o produto interno genérico da equação anterior pelo novo produto interno e, considerando-se o evento  $\vec{A}$  como um evento exclusivo ou incompatível num universo  $\vec{U}$ :

$$\vec{B}_{\vec{A}} = \frac{n(\vec{B} \cap \vec{A})}{n(\vec{A} \cap \vec{A})} \cdot \vec{A} = \frac{n(\vec{B} \cap \vec{A})}{n(\vec{A})} \cdot \vec{A} = P(\vec{B} / \vec{A}) \cdot \vec{A} \quad (39)$$

Foi possível substituir  $n(\vec{A} \cap \vec{A})$  por  $n(\vec{A})$  no denominador por causa de supor que  $\vec{A}$  é um evento exclusivo ou excludente que permite que  $\vec{A} \cap \vec{A} = \vec{A}$ . Neste

caso, poderia-se considerar que o universo  $\bar{u}$  está constituído por dois eventos que excluem-se mutuamente, o evento  $\bar{A}$  e o evento ortogono-complementar  $\bar{\bar{A}}$ .

Da anterior expressão pode-se obter:

$$\frac{n(\bar{B} \cap \bar{A})}{n(\bar{A} \cap \bar{A})} = n(\bar{B} \cap \frac{\bar{A}}{n(\bar{A})}) = n(\bar{B} \cap \bar{a})$$

$$n(\bar{B} \cap \bar{a}) = P(\bar{B} / \bar{A}) \quad (40)$$

desta forma o produto interno de  $\bar{B}$ , pela normalização de  $\bar{A}, \bar{a}$ , fornece a probabilidade condicionada de  $\bar{B}$  tendo previamente  $\bar{A}$ .

Calcularemos agora a probabilidade da projeção de  $\bar{B}_{\bar{A}} = n(\bar{B} \cap \bar{a}) \cdot \bar{A}$  a qual é muito útil na tarefa de reconhecimento de padrões:

$$P(\bar{B}_{\bar{A}}) = n(\bar{B}_{\bar{A}} \cap \bar{u}) = n(n(\bar{B} \cap \bar{a}) \cdot \bar{A} \cap \bar{u}) =$$

$$= n(\bar{B} \cap \bar{a}) \cdot n(\bar{A} \cap \bar{u}) = P(\bar{B} / \bar{A}) \cdot P(\bar{A})$$

$$P(\bar{B}_{\bar{A}}) = P(\bar{B} / \bar{A}) \cdot P(\bar{A}) \quad (41)$$

Ao invés de  $\bar{B}$  considere-se um outro vetor  $\bar{X}$  formado por uma determinada combinação de atributos (seus componentes). Gostaríamos de classificar o vetor  $\bar{X}$  de acordo com sua proximidade a outros padrões  $\bar{A}, \bar{B}, \bar{C}, \dots$  sendo cada um destes padrões representantes da sua própria classe de padrões. A maior das projeções  $\bar{A}_{\bar{X}}, \bar{B}_{\bar{X}}, \bar{C}_{\bar{X}}, \dots$  determinará qual dos padrões  $\bar{A}, \bar{B}, \bar{C}, \dots$  é mais similar ao  $\bar{X}$ . Uma maneira de medir a projeção e decidir qual é a maior é projetando-se sobre  $\bar{u}$  as projeções anteriores e obtendo-se assim  $P(\bar{A}_{\bar{X}}), P(\bar{B}_{\bar{X}}), P(\bar{C}_{\bar{X}}), \dots$  (vide equação 19). O maior destes valores (funções discriminantes na teoria clássica de reconhecimento de padrões) determinarão o padrão mais semelhante ao  $\bar{X}$ . O critério anterior é conhecido como a regra de decisão de Bayes [2,4,42]. Neste trabalho esta regra estatística foi

desenvolvida utilizando-se um novo ponto de vista a partir da álgebra linear.

Nesta seção, a seguinte expressão na qual utilizam-se dois vetores arbitrários  $\vec{X}$  e  $\vec{A}$ , será demonstrada:

$$P(\vec{A}_{\vec{X}}) = P(\vec{A} \cap \vec{X}) \quad (42)$$

Primeiro calcularemos a probabilidade da interseção para chegar, no final, ao primeiro membro da equação 42,

$$P(\vec{A} \cap \vec{X}) = n(\vec{A} \cap \vec{X} \cap \vec{u}) = n(\vec{A} \cap \vec{X} \cap \vec{u}) =$$

multiplicando-se e dividindo-se por  $n(\vec{X} \cap \vec{u})$ ,

$$\begin{aligned} n(\vec{X} \cap \vec{u}) \cdot n(\vec{A} \cap \frac{\vec{X} \cap \vec{u}}{n(\vec{X} \cap \vec{u})}) = \\ n(\vec{X} \cap \vec{u}) \cdot n(\vec{A} \cap \frac{(\vec{X} \cap \vec{U})/n(\vec{U})}{n(\vec{X} \cap \vec{U})/n(\vec{U})}) = \end{aligned}$$

e tendo em vista que  $\vec{X} \cap \vec{U} = \vec{X}$

$$\begin{aligned} &= n(\vec{X} \cap \vec{u}) \cdot n(\vec{A} \cap \frac{\vec{X}}{n(\vec{X})}) = \\ &= n(\vec{X} \cap \vec{u}) \cdot n(\vec{A} \cap \vec{x}) = P(\vec{X}) \cdot P(\vec{A} / \vec{X}) \end{aligned}$$

obtendo assim:

$$P(\vec{A} \cap \vec{X}) = P(\vec{X})P(\vec{A} / \vec{X}) = P(\vec{A}_{\vec{X}}) \quad (43)$$

a qual é a expressão que estávamos buscando. Uma expressão relacionada, que é o clássico teorema de Bayes deriva-se diretamente da expressão anterior:

$$P(\bar{A} / \bar{X}) = \frac{P(\bar{A} \cap \bar{X})}{P(\bar{X})} \quad (44)$$

a qual é a probabilidade de  $\bar{A}$  depois de ser fornecido o vetor  $\bar{X}$ .

Segundo a equação 43 a regra de decisão de Bayes (que permite classificar um vetor  $X$  segundo sua proximidade a outros vetores  $A, B, \text{ etc.}$ ) pode utilizar várias versões (todas equivalentes) da mesma função:  $P(\bar{A}_{\bar{X}}), P(\bar{X})P(\bar{A} / \bar{X})$  ou  $P(\bar{A} \cap \bar{X})$ . Estes tipos de funções são chamadas de funções discriminantes na área de reconhecimento de padrões [2, 4, 6]. No nosso caso, a última versão  $P(\bar{A} \cap \bar{X})$  será a mais freqüentemente utilizada.

### ***Eventos estatísticos independentes***

De acordo com a estatística clássica, um evento  $A$  é independente de um evento  $B$  se:

$$P(A / B) = P(A) \quad (45)$$

Analisemos esta fórmula, usando nosso conhecimento sobre produtos escalares. O primeiro membro da igualdade é

$$P(\bar{A} / \bar{B}) = \frac{n(\bar{A} \cap \bar{B})}{n(\bar{B})} = n(\bar{A} \cap \bar{b})$$

e o segundo membro:

$$P(\bar{A}) = n(\bar{A} \cap \bar{u})$$

Se  $\bar{b} = \bar{u}$ , então  $n(\bar{A} \cap \bar{b})$  (o valor da projeção do vetor  $\bar{A}$  sobre  $\bar{b}$ ) iguala a  $n(\bar{A} \cap \bar{u})$  (ou seja o valor da projeção do vetor  $\bar{A}$  sobre  $\bar{u}$ ). Isto significa que, do ponto de vista de  $\bar{A}$ , o universo  $\bar{U}$  e o vetor  $\bar{B}$  são equivalentes ou pelo menos co-lineares (seus módulos podem ser diferentes mas sua direção não). Resumindo a idéia sobre independência podemos formular o seguinte

teorema:

**Teorema de independência:**

Dois eventos  $\vec{A}$  e  $\vec{B}$  são independentes se o espaço amostral de um deles, por exemplo o espaço amostral de um evento  $\vec{A}$  é co-linear com o vetor  $\vec{B}$  ou vice-versa.

A condição  $n(\vec{A} \cap \vec{b}) = n(\vec{A} \cap \vec{u})$  é como se, de alguma forma, o vetor  $\vec{B}$  estivesse "colorindo" todo espaço do evento  $\vec{A}$ , de forma que o espaço amostral de  $\vec{A}$ , chamemo-lo de  $\vec{U}$ , torna-se um espaço amostral  $\vec{U}$  tipo-B. Da mesma forma, cada um dos eixos do espaço amostral  $\vec{U}$ , é "colorido" pelo evento  $\vec{B}$ , tendo um  $\vec{y}_1$  tipo- $\vec{B}$ , um  $\vec{y}_2$  tipo- $\vec{B}$ , etc. O próximo exemplo esclarecerá algumas dúvidas. Suponhamos que temos duas máquinas que trabalham continuamente e independentemente. Uma destas máquinas produz dois tipos de peças, por exemplo, peças grandes e peças pequenas; enquanto a outra máquina gera cores, por exemplo preto e branco. O output de ambas as máquinas é combinado na saída da máquina de forma que quatro tipos de combinações são geradas: grande-preto, grande-branco, pequeno-preto, pequeno-branco. Suponhamos que a máquina de cores produz preto 30% do tempo e produz branco 70% das vezes. Quando a máquina de cores está produzindo preto ou branco não se preocupa com o tamanho das peças, de forma que o preto está colorindo 30% das peças, sem importar se são grandes ou pequenas.

$$\begin{aligned} \overrightarrow{\text{Pr eto}} &= n(\overrightarrow{\text{Pr eto}} \cap \overrightarrow{\text{grande}})\overrightarrow{\text{Grande}} + n(\overrightarrow{\text{Pr eto}} \cap \overrightarrow{\text{pequeno}})\overrightarrow{\text{Pequeno}} = \\ &0.3\overrightarrow{\text{Grande}} + 0.3\overrightarrow{\text{Pequeno}} = 0.3(\overrightarrow{\text{Grande}} + \overrightarrow{\text{Pequeno}}) = 0.3\vec{T} \end{aligned} \tag{46}$$

Onde nós chamamos o espaço amostral de tamanhos  $\vec{T}$ , e seu vetor normalizado  $\vec{i}$ . Demonstraremos agora que  $P(\overrightarrow{Grande} / \overrightarrow{Preto}) = P(\overrightarrow{Grande})$  o qual é a condição para a independência dos eventos  $\overrightarrow{Grande}$  e  $\overrightarrow{Preto}$ :

$$P(\overrightarrow{Grande} / \overrightarrow{Preto}) = \frac{n(\overrightarrow{Grande} \cap \overrightarrow{preto})}{n(\overrightarrow{preto})} = \frac{n(\overrightarrow{Grande} \cap \frac{0.3\vec{T}}{n(0.3\vec{T})})}{n(0.3\vec{T})} = \frac{n(\overrightarrow{Grande} \cap \vec{i})}{n(\vec{i})} = P(\overrightarrow{Grande}) \quad (47)$$

Neste exemplo o espaço amostral de tamanhos é colorido de duas formas. Isto é, pode ser preto ou branco: 30% dos tamanhos são preto e 70% deles é branco. Estas proporções não variam, apesar de que se considerem peças grandes ou pequenas.

Cada um destes dois espaços amostrais, o espaço amostral de tamanhos,  $\vec{T}$ , e o espaço amostral de cores,  $\vec{c}$ , podem ser considerados sub-espacos amostrais mutuamente independentes.

### ***Transformações lineares em espaços Euclidianos probabilísticos.***

O seguinte exemplo introduz a representação matricial nestes espaços Euclidianos-probabilísticos, e além disto, demonstra que a estatística Euclidiana permite resolver problemas tradicionais de estatística de um modo bastante simples:

Sejam duas urnas. A urna 1 contendo  $x$  bolas brancas e  $y$  bolas vermelhas. A urna 2 contem  $z$  bolas brancas e  $v$  bolas vermelhas. Uma bola é escolhida ao acaso da urna 1 e posta na urna 2. A seguir, uma bola é escolhida ao acaso da urna 2. Qual será a probabilidade de que esta bola seja branca e qual a probabilidade de que seja vermelha?

Este problema pode ser resolvido como uma transformação linear de uma base de eixos ortogonais em uma outra base de eixos ortogonais. No momento da extração da primeira bola só se pode obter uma bola branca ou uma vermelha, excluindo-se mutuamente estes dois eventos, pode-se então considerar que estes eventos constituem uma primeira base de eixos ortogonais. Do mesmo modo, os eventos: escolher ao acaso uma bola branca ou escolher uma bola vermelha, da segunda urna constituem uma segunda base de eixos ortogonais.

Para abreviar a notação chamaremos de  $\bar{b}$  ao evento extrair uma bola branca e de  $\bar{v}$  ao evento extrair uma bola vermelha. U1 significará urna 1 e U2 urna 2. Em conseqüência  $\bar{b}_{U1}$  pode ser traduzido como o evento que consiste em

tirar uma bola branca da primeira urna.

As probabilidades de cada eixo da primeira base são calculados da seguinte maneira:

$$P(\vec{B}_{U1}) = \frac{x}{x+y}$$

$$P(\vec{V}_{U1}) = \frac{y}{x+y}$$

Cada um dos eixos da segunda base podem ser obtidos como combinação linear dos eixos da primeira base assim:

$$\vec{B}_{U2} = P(\vec{B}_{U2} / \vec{B}_{U1}) \cdot \vec{B}_{U1} + P(\vec{B}_{U2} / \vec{V}_{U1}) \cdot \vec{V}_{U1}$$

$$\vec{V}_{U2} = P(\vec{V}_{U2} / \vec{B}_{U1}) \cdot \vec{B}_{U1} + P(\vec{V}_{U2} / \vec{V}_{U1}) \cdot \vec{V}_{U1}$$

Podem-se representar as equações anteriores como a seguinte transformação linear:

$$\begin{bmatrix} \vec{B}_{U2} \\ \vec{V}_{U2} \end{bmatrix} = \begin{bmatrix} P(\vec{B}_{U2} / \vec{B}_{U1}) & P(\vec{B}_{U2} / \vec{V}_{U1}) \\ P(\vec{V}_{U2} / \vec{B}_{U1}) & P(\vec{V}_{U2} / \vec{V}_{U1}) \end{bmatrix} \begin{bmatrix} \vec{B}_{U1} \\ \vec{V}_{U1} \end{bmatrix}$$

E calculando cada um dos valores da matriz :

$$\begin{bmatrix} \vec{B}_{U2} \\ \vec{V}_{U2} \end{bmatrix} = \begin{bmatrix} \frac{z+1}{z+v+1} & \frac{z}{z+v+1} \\ \frac{v}{z+v+1} & \frac{v+1}{z+v+1} \end{bmatrix} \begin{bmatrix} \vec{B}_{U1} \\ \vec{V}_{U1} \end{bmatrix}$$

Calculando-se o produto interno dos eixos da segunda base pelo vetor

unitário  $\bar{u}$  obtemos as probabilidades  $P(\bar{B}_{U_2})$  e  $P(\bar{V}_{U_2})$ :

$$\begin{bmatrix} P(\bar{B}_{U_2}) \\ P(\bar{V}_{U_2}) \end{bmatrix} = \begin{bmatrix} P(\bar{B}_{U_2} / \bar{B}_{U_1}) & P(\bar{B}_{U_2} / \bar{V}_{U_1}) \\ P(\bar{B}_{U_2} / \bar{B}_{U_1}) & P(\bar{B}_{U_2} / \bar{V}_{U_1}) \end{bmatrix} \begin{bmatrix} P(\bar{B}_{U_1}) \\ P(\bar{V}_{U_1}) \end{bmatrix}$$

Que, finalmente, fornece o resultado seguinte:

$$\begin{bmatrix} P(\bar{B}_{U_2}) \\ P(\bar{V}_{U_2}) \end{bmatrix} = \begin{bmatrix} \frac{z+1}{z+v+1} & \frac{z}{z+v+1} \\ \frac{v}{z+v+1} & \frac{v+1}{z+v+1} \end{bmatrix} \begin{bmatrix} x \\ x+y \\ y \\ x+y \end{bmatrix}$$

Na segunda e terceira parte desta tese, este tipo de transformações matriciais serão aplicadas nas áreas da lógica *fuzzy* e às redes neurais. Ambas disciplinas permitem um mapeamento desde um espaço de entrada a um espaço de saída. Demonstrar-se-á que, na lógica *fuzzy* este tipo de matriz de transformação linear permite implementar um conjunto de regras do tipo *if-then* enquanto que as redes neurais que serão apresentadas na Parte III fazem um mapeamento desde um espaço de dimensão elevada a um espaço de dimensão pequena, permitindo uma extração de atributos a partir dos padrões de entrada.

***Uma versão estatística do algoritmo de Gram-Schmidt.***

Até agora vimos várias expressões úteis para classificação de padrões como a regra de decisão de Bayes e a probabilidade da interseção de padrões. Contudo, antes de realizarmos a classificação de padrões, a tarefa de extração dos atributos é inevitável. Neste caso, esta tarefa pode ser feita mediante o processo iterativo de Gram-Schmidt. Este processo permite extrair os atributos ortogonais (eixos)  $\vec{Y}_i$  a partir de um conjunto de padrões  $\vec{B}_{r+1}$  não necessariamente ortogonais. Neste algoritmo  $G_i$  significa a iteração  $i$ -ésima de Gram-Schmidt. A expressão genérica do algoritmo de Gram-Schmidt é a seguinte:

$$\begin{aligned} \vec{Y}_1 &= G_1(\vec{B}_1) = \vec{B}_1 \\ \vec{Y}_{r+1} &= G_{r+1}(\vec{B}_{r+1}) = \vec{B}_{r+1} - \sum_{i=1}^r \frac{(\vec{B}_{r+1}, \vec{Y}_i)}{(\vec{Y}_i, \vec{Y}_i)} \cdot \vec{Y}_i \\ &\text{para } r=1,2,\dots,k-1 \end{aligned} \tag{48}$$

Substituindo-se o novo produto interno  $(\vec{B}, \vec{y}_i) = n(\vec{B} \cap \vec{y}_i)$  na expressão acima, obtém-se os eixos  $\vec{Y}_i$  como segue:

$$\begin{aligned} \vec{Y}_1 &= \vec{B}_1 \\ \vec{Y}_{r+1} &= \vec{B}_{r+1} - \sum_{i=1}^r \left[ \frac{n(\vec{B}_{r+1} \cap \vec{Y}_i)}{n(\vec{Y}_i \cap \vec{Y}_i)} \right] \cdot \vec{Y}_i = \\ &\vec{B}_{r+1} - \sum_{i=1}^r n(\vec{B}_{r+1} \cap \vec{Y}_i) \cdot \vec{y}_i \end{aligned} \tag{49}$$

A expressão em colchetes é equivalente a  $P(\vec{B}_{r+1} / \vec{Y}_i)$ , de maneira que a expressão superior pode também ser escrita assim:

$$\begin{aligned}\vec{Y}_1 &= \vec{B}_1 \\ \vec{Y}_{r+1} &= \vec{B}_{r+1} - \sum_{i=1}^r [P(\vec{B}_{r+1} / \vec{Y}_i)] \vec{Y}_i\end{aligned}\quad (50)$$

ou:

$$\begin{aligned}\vec{Y}_1 &= \vec{B}_1 \\ \vec{Y}_{r+1} &= \vec{B}_{r+1} - \sum_{i=1}^r \vec{B}_{Y_i}\end{aligned}\quad (51)$$

Nestas equações cada vetor ortogonal  $\vec{Y}_i$  representa o atributo representativo no novo padrão que não pode ser obtido mediante uma combinação linear dos eixos encontrados previamente. Este conjunto de eixos  $\vec{Y}_i$  constituem uma base que permite gerar o mesmo sub-espço gerado pelos padrões  $\vec{B}_i$ . Aplicando-se a nova definição de probabilidade sob os dois membros da equação 50.

$$\begin{aligned}P(\vec{Y}_1) &= P(\vec{B}_1) \\ P(\vec{Y}_{r+1}) &= P(\vec{B}_{r+1}) - \sum_{i=1}^n P(\vec{B}_{r+1} / \vec{Y}_i) \cdot P(\vec{Y}_i)\end{aligned}\quad (52)$$

Calculemos agora o resultado de aplicar Gram-Schmidt a um padrão  $\vec{B}_{r+1}$  formado pela soma de outros dois  $\vec{B}_{r+1} = \vec{B}_{r+1}^1 + \vec{B}_{r+1}^2$ . Aplicamos, por exemplo a segunda iteração de Gram-Schmidt, independentemente, a cada um dos dois sub padrões e, depois, calculamos a soma dos dois resultados. Pode-se ver que o resultado é o mesmo que aplicar-se diretamente Gram-Schmidt à soma dos dois sub padrões:

$$G_{i=2}(\vec{B}_{r+1}^1) + G_{i=2}(\vec{B}_{r+1}^2) = G_{i=2}(\vec{B}_{r+1}^1 + \vec{B}_{r+1}^2)\quad (53)$$

Segue-se a demonstração:

$$\begin{aligned}\bar{Y}_2^1 &= G_{i=2}(\bar{B}_2^1) = \bar{B}_2^1 - (\bar{B}_2^1 \cap \bar{Y}_1) \cdot \bar{y}_1 \\ \bar{Y}_2^2 &= G_{i=2}(\bar{B}_2^2) = \bar{B}_2^2 - (\bar{B}_2^2 \cap \bar{Y}_1) \cdot \bar{y}_1\end{aligned}$$

$$\begin{aligned}G_{i=2}(\bar{B}_2^1) + G_{i=2}(\bar{B}_2^2) &= \\ \left[ \bar{B}_2^1 - (\bar{B}_2^1 \cap \bar{Y}_1) \cdot \bar{y}_1 \right] + \left[ \bar{B}_2^2 - (\bar{B}_2^2 \cap \bar{Y}_1) \cdot \bar{y}_1 \right] &= \\ (\bar{B}_2^1 + \bar{B}_2^2) - ((\bar{B}_2^1 + \bar{B}_2^2) \cap \bar{Y}_1) \cdot \bar{y}_1 &= G_{i=2}(\bar{B}_2^1 + \bar{B}_2^2)\end{aligned} \quad (54)$$

Este resultado significa que as iterações das equações de Gram-Schmidt são aplicadas independentemente sobre  $\bar{B}_i^1, \bar{B}_i^2$  e depois, as contribuições parciais são somadas para obter  $\bar{Y}_i$ . O mesmo princípio de linearidade pode ser aplicado a um padrão que seja a soma de m vetores multiplicados, cada um, por um fator, assim:  $\sum_{j=1}^m K^j \bar{B}_{i+1}^j$ . Deste modo é possível generalizar este resultado da seguinte forma:

$$G_{i=k} \left( \sum_{j=1}^m \{ K^j \bar{B}^j \} \right) = \sum_{j=1}^m \{ K^j \cdot G_{i=k}(\bar{B}^j) \} \quad (55)$$

Tendo-se aplicado Gram-Schmidt a um padrão qualquer (simples ou composto), o resultado vai ser uma sucessão de n vetores ortogonais representados no espaço n-dimensional dos próprios vetores  $\bar{B}_i$ , sendo possível encontrar uma transformação linear para encontrar os componentes de qualquer vetor n-dimensional no novo espaço m-dimensional.

*Uma alternativa ao algoritmo de ortogonalização de Gram-Schmidt a fim de permitir a extração dos chamados componentes principais do conjunto de padrões.*

No algoritmo de Gram-Schmidt os vetores  $\vec{b}_i$  são fornecidos numa certa ordem que determina os valores e direções dos eixos gerados. Este fato representa uma importante limitação do método pois os primeiros vetores apresentados, que geram os primeiros eixos, poderiam ser vetores pouco representativos do conjunto total de vetores, ou incluso “outliers” [2, 4, 6], ou seja vetores afastados do resto. No algoritmo que será explicado a seguir não se tem a limitação anterior pois todos os vetores  $\vec{b}_i$  são considerados ao mesmo tempo.

Sugerimos ao leitor, antes de continuar, que lesse as considerações finais da seção “Eventos estatísticos considerados como vetores num espaço Euclidiano.” Embora aquelas considerações em relação à aceção ideal e populacional de um evento poderiam parecer vagas naquele momento, são de grande importância para a compreensão da seguinte expliação.

A seguir aplicaremos Gram-Schmidt, não a padrões  $\vec{P}_i$  concretos, e sim ao vetor genérico  $\vec{P}$  que representa um conjunto de padrões individuais.

Deste modo temos a primeira iteração de Gram-Schmidt:

$$\vec{Y}_1 = \vec{P}$$

Esto significa que o vetor  $Y_1$  equivale ao vetor  $\vec{P}$  (seja na sua aceção ideal

ou populacional). Na acepção populacional estaríamos ante um vetor que representa a média dentre todos os  $\vec{P}_i$ .

A segunda iteração de Gram-Schmidt seria:

$$\vec{Y}_2 = \vec{P} - \vec{P}_{\vec{Y}_1}$$

Na acepção populacional  $\vec{P}_{\vec{Y}_1}$  seria cada uma das projeções dos vetores  $\vec{P}_i$  sobre o vetor médio  $\vec{Y}_1$ . Estamos, portanto, tirando de cada  $\vec{P}_i$  (que pertencem ao conjunto  $\vec{P}$  de vetores) o componente desses vetores sobre o vetor médio  $\vec{Y}_1$ . Como resultado temos um novo conjunto de vetores ortogonais a  $\vec{Y}_1$ , o conjunto dos vetores  $\vec{Y}_2$ , que pertencem a um plano ortogonal ao  $\vec{Y}_1$ . Portanto, na acepção ideal, o  $\vec{Y}_2$  representaria uma espécie de desvio padrão de tipo vetorial, ou seja a média de um conjunto de desvios em relação a uma outra média  $\vec{Y}_1$ . Além da média dos desvios, poderíamos calcular a média dos desvios máximos que daria a direção de máxima variabilidade no plano ortogonal a  $\vec{Y}_1$ , ou seja, o chamado primeiro componente principal. Uma forma de realizar isto seria aplicando a cada desvio uma função não linear crescente (tipo sigmoide, ou logística) para destacar os componentes dos desvios que são maiores, assim:

$$\vec{Y}_2 = \Psi(\vec{P} - \vec{P}_{\vec{Y}_1})$$

Com um raciocínio similar poderíamos extrair os componentes principais restantes.

Em resumo, o algoritmo de obtenção de componentes principais que está sendo proposto na tese seria o seguinte:

$$\begin{aligned}\vec{Y}_1 &= \vec{P} \\ \vec{Y}_{r+1} &= \Psi\left(\vec{P} - \sum_{i=1}^r \vec{P}_{\vec{Y}_i}\right)\end{aligned}\quad (56)$$

No caso de se preferir componentes principais normalizados:

$$\begin{aligned}\vec{y}_1 &= \vec{P} / \text{norm}(\vec{P}) \\ \vec{Y}_{r+1} &= \frac{\Psi\left(\vec{P} - \sum_{i=1}^r \vec{P}_{\vec{Y}_i}\right)}{\text{norm}\left(\Psi\left(\vec{P} - \sum_{i=1}^r \vec{P}_{\vec{Y}_i}\right)\right)}\end{aligned}$$

Até o momento, todos os padrões, tanto os  $\vec{P}_i$  como os  $\vec{Y}_i$ , resultado da aplicação do algoritmo de Gram-Schmidt foram representados na mesma base de vetores, os vetores unitários que poderíamos chamar  $\vec{b}_i$ . Mas segundo vao se obtendo os diferentes componentes principais obtemos uma base mais reduzida de eixos sobre os quais representar os vetores  $\vec{P}_i$ . Assim a igualdade da primeira iteração de Gram-Schmidt poderia ser expressa também da seguinte maneira:

$$\vec{Y}_1 = P(\vec{Y}_1 / \vec{b}_1)\vec{b}_1 + P(\vec{Y}_1 / \vec{b}_2)\vec{b}_2 + P(\vec{Y}_1 / \vec{b}_3)\vec{b}_3 + \dots + P(\vec{Y}_1 / \vec{b}_n)\vec{b}_n$$

Como veremos na Parte III, cada uma das probabilidades condicionais poderia ser interpretada como os n pesos das n sinapses de um neurônio genérico.

***Comparação entre diferentes interpretações geométricas de probabilidade.***

O modelo *fuzzy*-estatístico Euclidiano, que é proposto neste artigo, tem algumas similaridades com o modelo proposto por Bruno de Finetti em seu livro "Theory of Probability". Ambos os modelos analisam eventos estatísticos através de um ponto de vista geométrico. Contudo, várias diferenças muito importantes devem ser destacadas. Bruno de Finetti [5, 7] define o produto interior em um espaço euclidiano de variáveis estatísticas X, como:

$$(X, Y) = P\{X - P(X) \mid Y - P(Y)\} \quad (57)$$

Nesta expressão, ele usa P, previsão ou probabilidade, dentro do produto interior. No nosso caso, a probabilidade não está dentro do produto interior mas é o resultado deste:  $P(\bar{X}) = (\bar{X}, \bar{u}) = n(\bar{X} \cap \bar{u})$ , onde  $n(\cdot)$  significa número de vezes, ou cardinalidade, e é definido como:

$$\bar{u} = \frac{\bar{U}}{n(\bar{U})} = \frac{\sum_{i=1}^n \bar{Y}_i}{n(\sum_{i=1}^n \bar{Y}_i)} = \frac{\sum_{i=1}^n \bar{Y}_i}{\sum_{i=1}^n n(\bar{Y}_i)} \quad (58)$$

Onde os  $\bar{Y}_i$  são a base dos eixos ortogonais  $n(\bar{Y}_i \cap \bar{Y}_j) = 0, \forall i \neq j$  que gera o espaço amostral ou universo  $\bar{U}$ .

De Finetti considera a probabilidade um caso especial de uma medida subjetiva e coerente, chamada previsão, onde as condições de coerência não impõem limites na atribuição arbitrária de probabilidades, com exceção de

que estas probabilidades não podem estar em contradição entre elas [5, p. 73]. Apesar de que nossa definição de produto escalar seja uma medida objetiva, o número de vezes que uma interseção de eventos acontece, pode ser também de natureza subjetiva, por exemplo o grau de superposição espaço-temporal subjetiva de dois eventos. As condições de coerência necessária e suficientes para esta medida, coincidem com as condições que caracterizam todo produto interior, ou seja, a comutatividade, distributividade, associatividade e positividade. Kosko, em seu livro "Neural Networks and *Fuzzy* Systems" [15] propõe um tipo de medida que inicialmente poderia ser utilizada nesta situação. Ele denomina esta medida "*fuzzy* count of overlap M", que será traduzido como grau de superposição de dois eventos *fuzzy*. Por exemplo, para dois conjuntos *fuzzy*:  $A=(1/3, 3/4)$  e  $B=(2/3, 3/4)$ , o grau de superposição entre eles é obtido da seguinte forma:

$$\begin{aligned} M(A \cap B) &= M(\{1/3, 3/4\} \cap \{2/3, 1/4\}) = \\ M(\min\{1/3, 2/3\} + \min\{3/4, 1/4\}) &= \\ M(1/3 + 1/4) &= 7/12 = 0,58 \end{aligned}$$

Contudo, o produto escalar mostrado neste artigo gera um resultado diferente:

$$\begin{aligned} n(\vec{A} \cap \vec{B}) &= n\left(\left\{\frac{1}{3}\vec{Y}_1 + \frac{3}{4}\vec{Y}_2\right\} \cap \left\{\frac{2}{3}\vec{Y}_1 + \frac{1}{4}\vec{Y}_2\right\}\right) = \\ n\left(\frac{1}{3} \cdot \frac{2}{3}\vec{Y}_1 + \frac{3}{4} \cdot \frac{1}{4}\vec{Y}_2\right) &= \frac{1}{3} \cdot \frac{2}{3}n(\vec{Y}_1) + \frac{3}{4} \cdot \frac{1}{4}n(\vec{Y}_2) \end{aligned}$$

Só no caso em que  $\vec{Y}_1$  e  $\vec{Y}_2$  estivessem normalizadas, ou seja  $n(\vec{Y}_1) = 1, n(\vec{Y}_2) = 1$  :

$$\begin{aligned}n(\bar{A} \cap \bar{B}) &= \frac{1}{3} \cdot \frac{2}{3} n(\bar{Y}_1) + \frac{3}{4} \cdot \frac{1}{4} n(\bar{Y}_2) = \\ &= \frac{1}{3} \cdot \frac{2}{3} + \frac{3}{4} \cdot \frac{1}{4} = 0,41\end{aligned}$$

## PARTE II. A LÓGICA FUZZY SOB A PERSPECTIVA DA ESTATÍSTICA EUCLIDIANA

### *Introdução*

Nesta segunda Parte utilizaremos os conceitos matemáticos explicados na Parte I para fundamentar a lógica *fuzzy*. A lógica *fuzzy* é uma ferramenta matemática que tem encontrado grande aplicação na área de controle de processos industriais. Inicialmente foi apresentada por seu criador Lotfi A. Zadeh como um linguagem para a interação homem-máquina [44]. Desde o primeiro momento encontrou defensores entusiastas e críticos acérrimos. Os críticos diziam que os fundamentos matemáticos da lógica *fuzzy* eram fracos e portanto a lógica *fuzzy* não deveria ser aceita como uma disciplina acadêmica rigorosa, enquanto os defensores argumentavam que a lógica *fuzzy* representava uma nova visão que abrangia a própria estatística e que tinha demonstrado-se superior às técnicas convencionais de controle. Este desacordo ainda perdura. Nas pesquisas que serão apresentadas nesta Parte II tentaremos suavizar estas discrepâncias. Na Parte I, vimos como a estatística poderia ser considerada como um ramo da álgebra linear pois um evento estatístico podia ser representado como um vetor. Estes eventos-vetores apresentados na Parte I equivalem, na lógica *fuzzy*, aos chamados conjuntos *fuzzy*. Portanto no nosso caso, falar de evento estatístico, evento-vetor ou conjunto *fuzzy* é exatamente o mesmo. Também foi visto na Parte I que a projeção de um evento-vetor (ou um conjunto *fuzzy*) sobre o vetor unitário,

que foi chamado  $\vec{u}$ , é uma medida que fornece a probabilidade deste evento ou conjunto *fuzzy*.

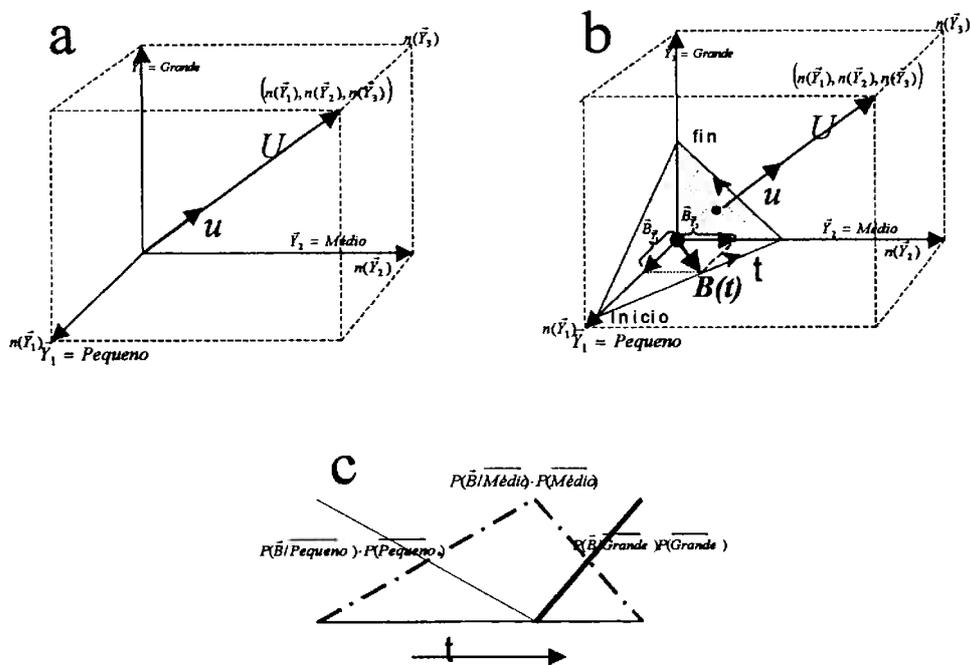
A Parte I terminava explicando que os componentes principais podem ser extraídos a partir de um conjunto de dados. Esses componentes principais constituíam a melhor base de eixos para representar um conjunto de dados, de forma que esses dados pudessem ser depois classificados ou utilizados para qualquer fim onde a variabilidade dos dados fosse importante.

Na lógica *fuzzy* também são procurados um conjunto de eixos que vão fornecer os componentes dos conjuntos *fuzzy* ou vetores eventos. Quando a lógica *fuzzy* é aplicada ao controle de um sistema, esta procura de eixos é feita entrevistando-se o operário que conhece o sistema. O operário identifica quais, segundo seu critério, são as variáveis que influem mais na dinâmica do sistema. Depois cada estado do sistema é descrito como um vetor no espaço das variáveis selecionadas. Esta seleção de variáveis também poderia ser feita encontrando-se os componentes principais a partir de um conjunto de dados extraídos da máquina ou sistema. Contudo esta extração de dados é, as vezes, difícil e demorada e, por este motivo, prefere-se em muitos casos, a experiência do operário ao invés da frieza dos dados. Acreditamos, embora não tenhamos dados para demonstrar a seguinte afirmação, que um bom operário termina achando subjetivamente, com a experiência, os componentes principais do sistema. Neste sentido, consideramos que o operário utiliza as redes neurais do seu próprio cérebro para obter esses componentes. Na terceira Parte demonstraremos como uma rede neural inspirada na biologia do ser humano pode executar esta extração dos componentes principais.

Além de procurar uns eixos, o desenhista de um sistema *fuzzy* procura a interligação dinâmica destas variáveis, ou seja como evoluem umas delas quando as outras são alteradas. Este conceito dá origem ao conceito de funções de pertinência. O desenho destas funções de pertinência parece, às vezes, arbitrário e de fato não existe unanimidade para responder a perguntas como: Por que as funções de pertinência são desenhadas como triângulos? Porque as funções de pertinência são desenhadas de maneira que em cada ponto a somatória dê um mesmo valor? etc. Começando por estas questões, tentaremos fundamentar nas seguintes seções, a lógica *fuzzy* dentro do contexto dos espaços Euclidianos de probabilidade os quais foram apresentados na Parte I.

**Funções de pertinência obtidas a partir de trajetórias em um plano.**

Imaginemos um espaço euclidiano tridimensional  $U^3$  no qual o vetor  $\vec{u}$  já apresentado na Parte I é representado (veja Figura 4.A). Representa-se também um plano P, com vetor característico  $\vec{u}$  que, por definição, é ortogonal a este plano ( Figura 4.B). Todos os vetores cujas origens estão no centro de coordenadas e cujas pontas das flechas estão neste plano, têm a mesma projeção sobre  $\vec{u}$ , e portanto a mesma probabilidade.



**Figura 4(a)** Somando-se todos os vetores da base de eixos coordenados e dividindo-se esta soma pela soma normalizada, o vetor unitário  $u$  é obtido. **(b)** Este vetor é normal ao plano P. Projetando-se qualquer vetor com a ponta da flecha no plano P sobre  $u$ , obtém-se a probabilidade de B. Por causa da definição de probabilidade, todos os vetores B, no plano P, têm a mesma probabilidade, assim como os vetores B da trajetória  $t$ . **(c)** A probabilidade dos componentes dos vetores cujas pontas das flechas estão nesta trajetória, são representadas na parte inferior do gráfico. Este gráfico é o mesmo que aquele que representa as funções de pertinência na lógica fuzzy.

Desenhemos uma trajetória  $t$  específica nas bordas deste plano. A direção de  $t$  está indicada com pontas de flecha acima da trajetória. As bordas correspondem a interseção do plano P com os planos ortogonais definidos pela combinação linear de vetores  $\vec{Y}_1$  e  $\vec{Y}_2$ ,  $\vec{Y}_2$  e  $\vec{Y}_3$ , e finalmente  $\vec{Y}_3$  e  $\vec{Y}_1$ . Neste exemplo, a trajetória  $t$  é descrita no sentido anti-horário do ponto etiquetado como “início” até o ponto etiquetado como “final”. As probabilidades de todos os vetores  $\vec{B}(t)$  cujas origens estão no centro de coordenadas e as pontas das flechas estão na trajetória  $t$ , serão representadas a seguir. Como todos os vetores  $\vec{B}(t)$  estão no plano P, todas as projeções de qualquer vetor  $\vec{B}(t)$  sobre  $\vec{u}$ , são iguais. Portanto todas as probabilidades  $P(\vec{B}(t))$  são as mesmas e correspondem à soma das probabilidades de cada componente destes vetores  $\vec{B}(t)$ . Cada componente dos vetores  $\vec{B}(t)$  é expresso da seguinte forma:  $P(\vec{B}(t)/\vec{Y}_i)\vec{y}_i$ . Projetando-se cada componente sobre o vetor unitário  $\vec{u}$  obteremos probabilidade de cada componente:

$$\begin{aligned} P(P(\vec{B}(t)/\vec{Y}_i)\vec{y}_i) &= n(P(\vec{B}(t)/\vec{Y}_i)\vec{Y}_i \cap \vec{u}) = \\ P(\vec{B}(t)/\vec{Y}_i) \cdot n(\vec{Y}_i \cap \vec{u}) &= P(\vec{B}(t)/\vec{Y}_i) \cdot P(\vec{Y}_i) \end{aligned} \quad (59)$$

Na Figura 4.C representam-se as probabilidades de cada componente em função do parâmetro  $t$ . Esta figura é a típica representação de função de pertinência *fuzzy* em função do valor de uma variável de entrada  $t$ .

O valor absoluto de  $P(\vec{B}(t))$  varia de universo para universo dependendo do

número de vetores  $\vec{B}(t)$ . Contudo, os valores relativos das probabilidades dos componentes de  $\vec{B}(t)$  são mais importantes que os valores absolutos. As probabilidades relativas de componentes de  $\vec{B}(t)$ , somente dependem da direção e sentido de  $\vec{B}(t)$  e não do módulo de  $\vec{B}(t)$  (que depende da distância do plano P ao origem de coordenadas) porque eles são influenciados igualmente quando  $\vec{B}(t)$  é multiplicado por qualquer fator k.

$$P(k \cdot \vec{B}) = n \left( \sum_{i=1}^n k \cdot \vec{B}_{\vec{Y}_i} \cap \vec{u} \right) \quad (60)$$

Como é mostrado na Figura 4.C, a soma dos componentes relativos é a mesma para todos os vetores  $\vec{B}(t)$  ao longo da trajetória t. Em outras palavras, a soma de todos os valores de pertinência é a mesma para cada vetor *fuzzy* ao longo da trajetória t, que é uma propriedade da lógica *fuzzy* amplamente utilizada na maior parte das implementações.

### ***Maximização da entropia e funções de pertinência.***

De acordo com a Teoria da Informação [36], a maximização de entropia em um sistema físico é equivalente a ter-se iguais probabilidades para cada ocorrência possível neste sistema. A teoria da Informação também destaca que a maximização de entropia corresponde a um estado de máxima informação fornecida pelo sistema. Estas idéias estão de acordo com o princípio da indiferença e também, com o princípio de máxima entropia de Jaynes [11] o qual amplia o princípio da indiferença. O princípio da indiferença afirma que, se a única informação sobre um experimento é que existem  $n$  eventos mutuamente excluintes, então a probabilidade de qualquer um destes eventos é  $1/n$ . Jaynes extrapola este princípio da seguinte forma: Se além da informação considerada no princípio de indiferença existe alguma informação adicional sobre um experimento, então as probabilidades não podem ser todas  $1/n$  e a distribuição mais adequada das probabilidades é a mais igualitária, tendo-se em conta a informação adicional. Este princípio de determinação de probabilidades é chamado por Jaynes, o princípio da máxima entropia.

Se a entropia é definida por:

$$h = -\sum_{i=1}^n p_i \ln p_i \quad (61)$$

Pode ser mostrado que a entropia é máxima, quando  $p_i=1/n$ , para todo  $i$ . Portanto o princípio de máxima entropia afirma que, tendo-se em conta a afirmação adicional, o que implicaria que as probabilidades não possam ser todas iguais a  $1/n$ , a distribuição mais adequada de probabilidades é aquela

que maximiza  $h$ .

Os princípios acima se aplicam ao exemplo da Figura 4. Neste caso, todos os vetores  $\vec{B}(t)$ , com origem no centro das coordenadas e com a ponta das flechas em um plano  $P$ , têm probabilidades idênticas. Portanto todos os pontos de qualquer trajetória descrita neste plano têm a mesma probabilidade. Esta probabilidade constante é a soma das funções de pertinência ou, em outras palavras, é a soma das probabilidades dos componentes em cada ponto da trajetória  $t$ . Esta somatória constante das funções de pertinência, constitui um dos métodos empíricos mais aplicados para designar funções de pertinência. Outro método, consiste no uso de funções de pertinência triangular. As funções de pertinência triangular, cujo uso é bastante difundido, permitem checar de forma rápida e fácil, a realização da limitação da somatória constante, que é uma condição imposta pelo princípio de maximização da entropia. Desta forma, um desenhador de funções de pertinência, está intuitivamente aplicando o princípio de máxima entropia de Jaynes, ao fixar um valor constante para a somatória das funções de pertinência.

Na Figura 4 a trajetória  $t$  no plano  $P$  gera uma função de pertinência triangular, apesar de que existam inúmeras outras possíveis trajetórias, que cumprem com o princípio acima como será mostrado em seguida. O fato de levar-se em consideração a maximização de informação, proporciona uma melhor performance quando aplicam-se as declarações condicionais da lógica *fuzzy*. Alguns pesquisadores como Kosko [15, p.277] e Terano *et al.*[40] também estudaram o conceito de entropia no contexto da lógica *fuzzy*. Contudo eles só apresentam este conceito isoladamente, sem um objetivo em

particular. Kosko define uma medida de entropia *fuzzy* para cada evento *fuzzy* da seguinte forma:

$$H(A) = \frac{M(A \cap \bar{A})}{M(A \cup \bar{A})} \quad (62)$$

Onde M, é o operador de cardinalidade que soma os componentes dos eventos *fuzzy*, e  $\bar{A}$  é o evento complementar do A.

Segundo Terano et al.[40], a “entropia da ambigüidade do conjuntos *fuzzy* A”, H(A) é descrita como:

$$H(A) = -k \sum_{i=1}^n \{ \mu_A(w_i) \log \mu_A(w_i) + \mu_{\bar{A}}(w_i) \log \mu_{\bar{A}}(w_i) \} \quad (63)$$

Ao invés disto, no nosso caso, aplicar-se-á a definição padrão de entropia diferencial h, ao longo da trajetória de pontos operacionais :

$$h = - \int_{start}^{end} f(t) \log f(t) dt = -E[\log f(t)] \quad (64)$$

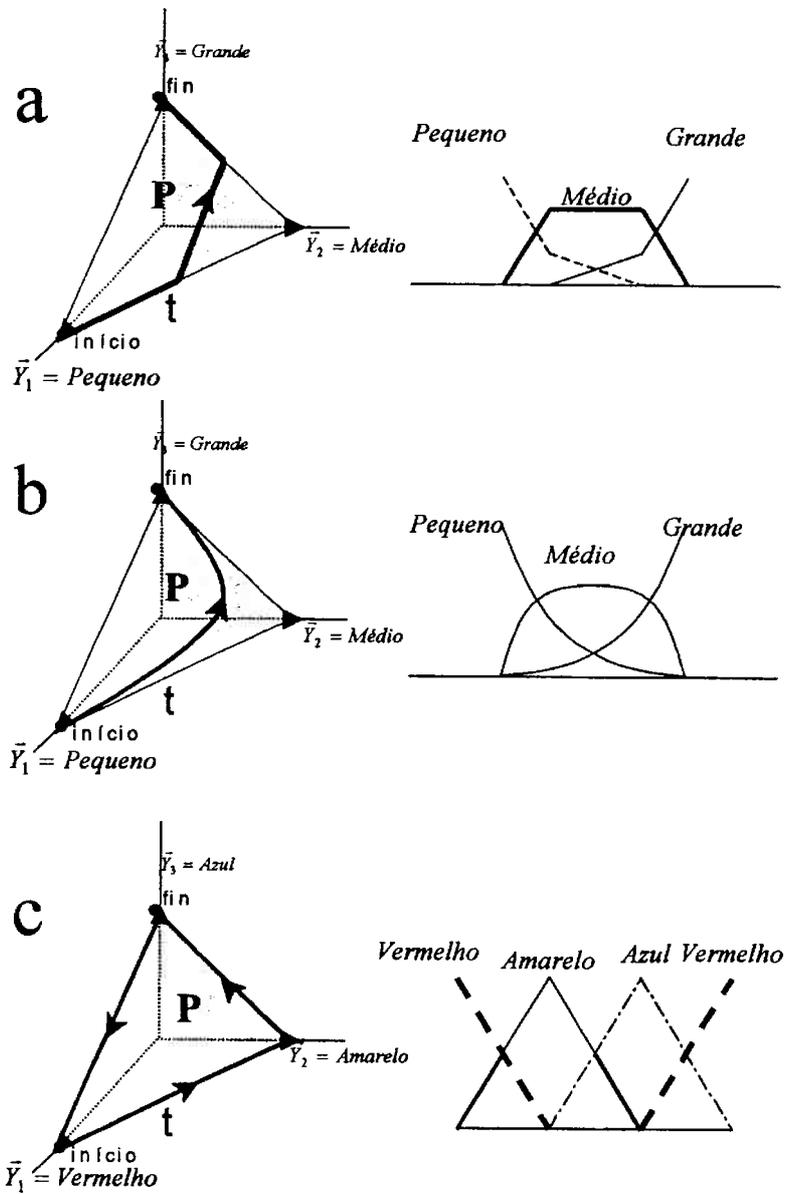
Onde f(t), é a função de densidade de probabilidade.

Considerando-se que  $f(t) = \sum_{i=1}^m \mu_i(t)$ , onde  $\mu_i(t)$  representa o valor de cada função de pertinência no ponto t, a entropia diferencial é:

$$h = - \int_{start}^{end} \sum_{i=1}^m \mu_i(t) \log \left( \sum_{i=1}^m \mu_i(t) \right) dt = -E \left[ \log \left( \sum_{i=1}^m \mu_i(t) \right) \right] \quad (65)$$

Levando-se em consideração a restrição de ter-se iguais probabilidades nos diferentes pontos operacionais das trajetórias, ou seja, a trajetória necessita pertencer ao plano P de equi-probabilidade, outras funções de pertinência são possíveis. Nestes outros casos, as funções de pertinência não se sobrepõem de dois em dois, podendo-se sobrepor 3 ou mais funções de pertinência. Alguns exemplos de função de pertinência alternativas são mostradas na

Figura 5.



**Figura 5** Nesta figura são mostradas diferentes formas de se obter um conjunto de funções de pertinência a partir de várias trajetórias desenhadas no plano de iso-probabilidade P. (a) Exemplo de função de pertinência trapezoidal. (b) Exemplo de função de pertinência curva. (c) Função de pertinência cíclico-triangular.

Na Figura 5.a, a trajetória descrita sobre o plano P, gera uma função de pertinência trapezoidal. Note que a altura das funções de pertinência não são iguais dependendo do fato ter mais de duas funções de pertinência sobrepondo-se em algum intervalo. Contudo a soma constante das funções de pertinência é conservada. Na Figura 5.c estão representadas uma trajetória fechada em P, e sua função de pertinência correspondente. Note que a função de pertinência para vermelho, é dividida em duas metades de triângulos no início, e no final da trajetória. Este tipo de situação não teria acontecido, se, ao invés da cor, o tamanho houvesse sido considerado, porque as categorias "pequena" e "grande" não podem ser diretamente compostas sem uma categoria "média". No caso das cores, as três categorias podem ser compostas em ordem cíclica.

### Entropia e desenho de funções de pertinência.

As considerações acima não são feitas unicamente com propósito especulativo. Estas considerações, auxiliam no desenho da forma das funções de pertinência. Por exemplo, se a tarefa é controlar a potência de uma máquina em um range de pequeno, médio e grande, e, se a magnitude for média na maioria dos casos, então o desenho da função de pertinência será como na Figura 6, tendo o vetor  $\vec{Y}_2$  um maior tamanho  $n(\vec{Y}_2)$ .

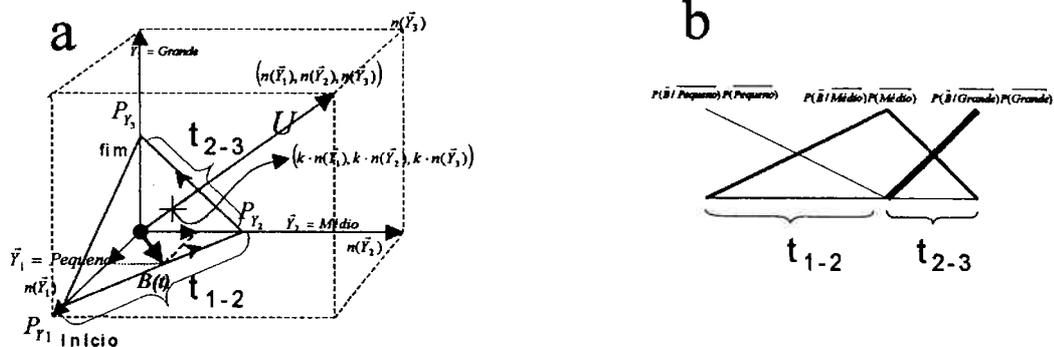


Figura 6(a) As distâncias  $t_{1-2}$  e  $t_{2-3}$  numa determinada trajetória contida no plano P, têm suas representações correspondentes, na função de pertinência do diagrama abaixo. (b) Desta maneira é possível obter-se a forma de uma função de pertinência a partir da trajetória desenhada no plano P.

Na Figura 6, a trajetória  $t$ , é dividida em dois seguimentos  $t_{1-2}$  e  $t_{2-3}$ . Usando-se a fórmula de Pitágoras, cada um destes valores são calculados:

$$t_{1-2} = \sqrt{P_{Y_1}^2 + P_{Y_2}^2}$$

$$t_{2-3} = \sqrt{P_{Y_2}^2 + P_{Y_3}^2} \quad (66)$$

Onde,  $P_{Y_1}$ ,  $P_{Y_2}$  e  $P_{Y_3}$  representam as medidas das distâncias que vão do centro das coordenadas aos pontos em que o plano P intersecta os diferentes eixos.

Estas distâncias são obtidas como segue:

Primeiro obteremos a equação do plano P com vetor característico  $(n(\vec{Y}_1), n(\vec{Y}_2), n(\vec{Y}_3))$ , que passa através do ponto  $(k \cdot n(\vec{Y}_1), k \cdot n(\vec{Y}_2), k \cdot n(\vec{Y}_3))$ .

$$n(\vec{Y}_1) \cdot (x - kn(\vec{Y}_1)) + n(\vec{Y}_2) \cdot (y - kn(\vec{Y}_2)) + n(\vec{Y}_3) \cdot (z - kn(\vec{Y}_3)) = 0 \quad (67)$$

Intersectando-se este plano P, com os eixos coordenados  $\vec{Y}_1, \vec{Y}_2, \vec{Y}_3$  são obtidas as distâncias do centro das coordenadas aos pontos de interseção:  $P_{y1}, P_{y2}, P_{y3}$ .

$$\begin{aligned} P_{Y_1} &= k \frac{n(\vec{Y}_1)^2 + n(\vec{Y}_2)^2 + n(\vec{Y}_3)^2}{n(\vec{Y}_1)} \\ P_{Y_2} &= k \frac{n(\vec{Y}_1)^2 + n(\vec{Y}_2)^2 + n(\vec{Y}_3)^2}{n(\vec{Y}_2)} \\ P_{Y_3} &= k \frac{n(\vec{Y}_1)^2 + n(\vec{Y}_2)^2 + n(\vec{Y}_3)^2}{n(\vec{Y}_3)} \end{aligned} \quad (68)$$

A fim de quantificar os valores  $n(\vec{Y}_i)$ , número de vezes que a magnitude é grande, média ou pequena, podemos usar a medida do intervalo de tempo que um técnico usa esta escala de magnitude durante sua tarefa. No próximo exemplo pretende-se ajudar no entendimento das idéias acima. Também será mostrado porque as funções de pertinência da esquerda e da direita, são desenhadas na maioria das implementações, com um lado horizontal como o representado na Figura 8. Também será mostrado como implementar uma função de pertinência trapezoidal, como o representado na Figura 5.a.

*Exemplo:*

Várias medidas na escala de 2 a 12 são apresentadas a um especialista. Ele decide classificar as medidas em três categorias : grande ( $\vec{Y}_1$ ), médio ( $\vec{Y}_2$ ) e pequeno ( $\vec{Y}_3$ ). É possível que este especialista decida que uma determinada medida é, por exemplo, metade média e metade pequena. Após ter feito esta classificação, o número de medidas pertencentes a cada uma das categorias é:  $n(\vec{Y}_1) = 15$ ,  $n(\vec{Y}_2) = 29$ ,  $n(\vec{Y}_3) = 9.5$ . Nos seguintes parágrafos responderemos as seguintes questões:

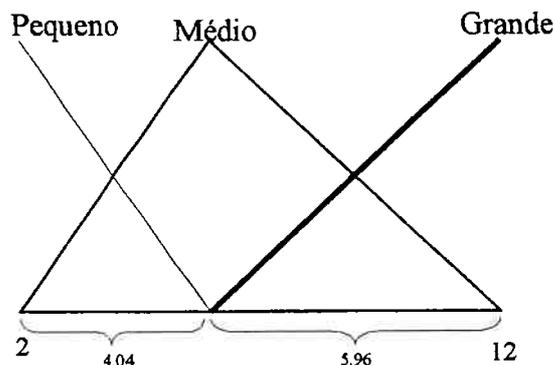
- a) De acordo com o princípio da máxima entropia, e sem ter nenhuma informação adicional sobre as medidas, além das acima descritas, qual seria a atribuição das funções de pertinência triangular?
- b) Qual seria a atribuição das funções de pertinência triangular se ao invés de três categorias, fossem consideradas cinco, sendo estas duas categorias adicionais denominadas : categoria muito grande ( $\vec{Y}_0$ ) e categoria muito pequena ( $\vec{Y}_4$ )? Neste caso particular estas duas novas categorias estariam intencionalmente vazias, ou seja :  $n(\vec{Y}_0) = 0$ ,  $n(\vec{Y}_4) = 0$ .
- c) Dê um exemplo de função de pertinência trapezoidal, pertencente ao plano P de iso-probabilidade.

Os valores acima permitem calcular  $P_{Y_1}$ ,  $P_{Y_2}$  e  $P_{Y_3}$  :  $P_{Y_1} = k*77.08$ ,  $P_{Y_2} = k*39.87$  and  $P_{Y_3} = k*121.71$

E finalmente:

$$t_{1,2}=k*86.78 \text{ e } t_{2,3}=k*128.07$$

Sabendo-se que o range da magnitude vai de 2 à 12, determinamos um valor para  $k=10/(86.78+128.07)$  de forma que as funções de pertinência são representadas como na Figura 7.



**Figura 7** Funções de pertinência desenhadas para uma situação na qual as únicas informações disponíveis são (a) que o domínio da variável tamanho vai, por exemplo, de 2 à 12, e (b) o número de medidas pertencentes as três diferentes categorias são  $n(\text{grande})=15$ ,  $n(\text{Média})=29$  e  $n(\text{Pequena})=9.5$ . Usando-se a informação geométrica dada pela fig. 6.a  $t_{1,2}$  e  $t_{2,3}$  são obtidos:  $t_{1,2}=4.04$  e  $t_{2,3}=5.96$ .

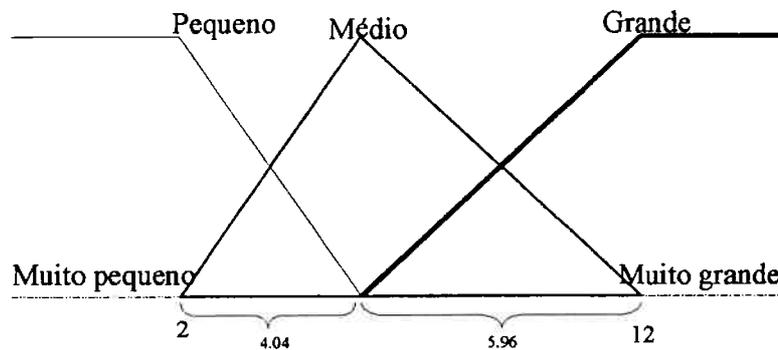
No caso de se usar cinco categorias: muito pequena, pequena, média, grande e muito grande, tendo-se as duas categorias adicionais vazias: muito pequena e muito grande, torna-se necessário calcular-se a interseção de um plano penta-dimensional com uma base de eixos penta-dimensional. A distância  $P_{y_i}$  entre cada uma destas interseções e a origem das coordenadas são (usando-se o valor previamente calculado para  $k$ ):

$$\begin{aligned}
P_{Y_0} &= k \frac{n(\bar{Y}_0)^2 + n(\bar{Y}_1)^2 + n(\bar{Y}_2)^2 + n(\bar{Y}_3)^2 + n(\bar{Y}_4)^2}{n(\bar{Y}_0)} = \\
&= k \frac{0^2 + 15^2 + 29^2 + 9.5^2 + 0^2}{0} = k \frac{1156.25}{0} = \infty \\
P_{Y_1} &= k \frac{1156.25}{n(\bar{Y}_1)} = k \frac{1156.25}{15} = k * 77.08 = 3.59 \\
P_{Y_2} &= k \frac{1156.25}{n(\bar{Y}_2)} = k * 39.87 = 1.86 \\
P_{Y_3} &= k \frac{1156.25}{n(\bar{Y}_3)} = k * 121.71 = 5.66 \\
P_{Y_4} &= k \frac{1156.25}{n(\bar{Y}_4)} = k \frac{1156.25}{0} = \infty
\end{aligned}$$

A partir dos valores acima, novos valores para  $t_{i,j}$  são obtidos:

$$\begin{aligned}
t_{0-1} &= \sqrt{P_{Y_0}^2 + P_{Y_1}^2} = \sqrt{\infty^2 + (3.59)^2} = \infty \\
t_{1-2} &= \sqrt{P_{Y_1}^2 + P_{Y_2}^2} = \sqrt{(3.59)^2 + (1.86)^2} = 4.04 \\
t_{2-3} &= \sqrt{P_{Y_2}^2 + P_{Y_3}^2} = \sqrt{(1.86)^2 + (5.66)^2} = 5.96 \\
t_{3-4} &= k \sqrt{P_{Y_3}^2 + P_{Y_4}^2} = k \sqrt{(5.66)^2 + \infty^2} = \infty
\end{aligned}$$

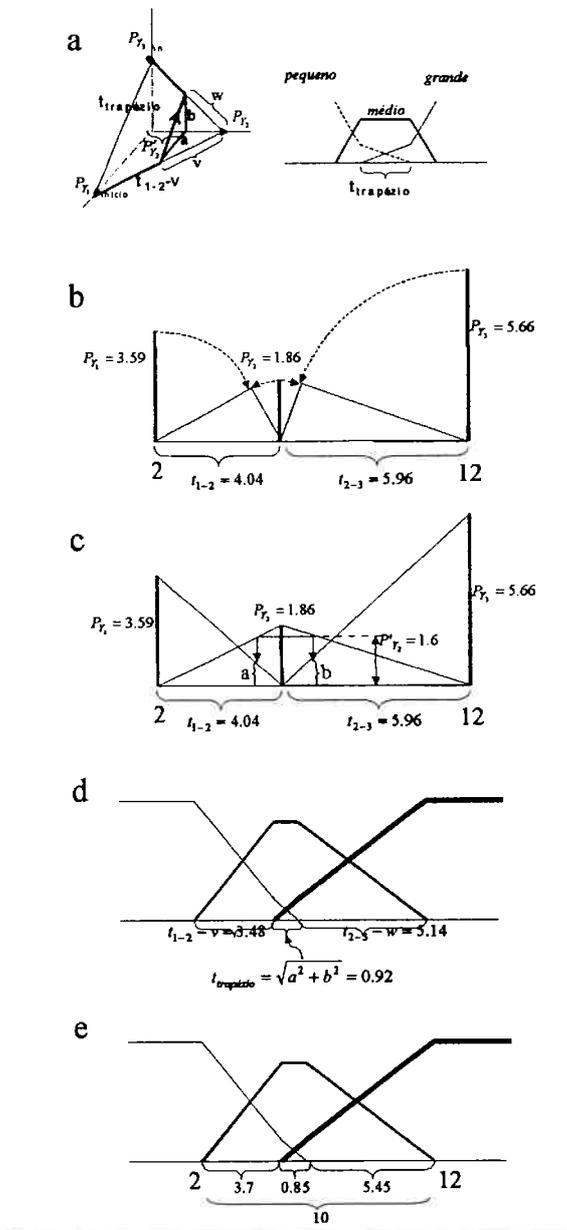
de forma que o diagrama de pertinência na Figura 8, é obtido:



**Figura 8** É mostrado o mesmo caso da fig.7 mas acrescentando-se duas novas categorias vazias: a categoria muito pequena e a categoria muito grande. Estas novas categorias produzem os novos componentes laterais horizontais desta figura.

Neste exemplo é mostrado que as funções de pertinência usuais direita e esquerda dos diagramas de pertinência, são na verdade, triângulos com uma base infinita.

Uma função de pertinência trapezoidal é obtida usando-se uma trajetória desenhada sobre um plano P (Figura 9.a).



**Figura 9** (a) Desenho de uma função trapezoidal (direita) é obtida a partir das propriedades geométricas do plano à esquerda. (b) As distâncias  $t_{1-2}$  e  $t_{2-3}$  são as hipotenusas dos triângulos, com lados  $P_{Y1}$ ,  $P_{Y2}$  e  $P_{Y2}$ ,  $P_{Y3}$  respectivamente. (c) Usando-se a similaridade dos triângulos é possível obter-se as distâncias  $a$ ,  $b$ ,  $v$ , e  $w$  que aparecem na figura 3-D acima. (d) A forma trapezoidal é obtida pela modificação da forma triangular (fig. 8) substituindo-se a parte central pela banda cuja largura é  $t_{trapezium}$ . (e) Finalmente o gráfico trapezoidal é normalizado, de forma que o domínio da

variável esteja entre 2 e 12.

O segmento superior do trapézio corresponde ao segmento da projeção constante sobre o eixo  $\bar{v}_2$ , na mesma figura. Neste caso um valor arbitrário de  $P'_{Y_2} = 1.6$  é dado para esta projeção constante. O comprimento do segmento superior do trapézio,  $t_{\text{trapézio}}$  é obtido deste valor  $P'_{Y_2}$ , através da derivação dos valores a e b, e a partir destes valores a e b obteremos  $t_{\text{trapézio}}$  da seguinte maneira:

$$t_{\text{trapézio}} = \sqrt{a^2 + b^2} \quad (69)$$

Os valores a e b são funções lineares das trajetórias  $t_{1,2}$  e  $t_{2,3}$ . O valor de a e b é zero quando  $P'_{Y_2} = P_{Y_2}$ . Quando  $P'_{Y_2} = 0$ , então a e b são respectivamente iguais a  $P_{Y_1}$  e  $P_{Y_3}$ .

As distâncias  $P_Y$  são representadas na Figura 9.b como linhas verticais, sendo  $t_{1,2}$  e  $t_{2,3}$ , as hipotenusas dos dois triângulos nesta figura.

As distâncias a e b são obtidas usando-se a propriedade de similaridade dos triângulos representadas na figura c :

$$\begin{aligned} a &= \frac{(P_{Y_2} - P'_{Y_2})P_{Y_1}}{P_{Y_2}} = 0.5 \\ b &= \frac{(P_{Y_2} - P'_{Y_2})P_{Y_3}}{P_{Y_2}} = 0.78 \end{aligned} \quad (70)$$

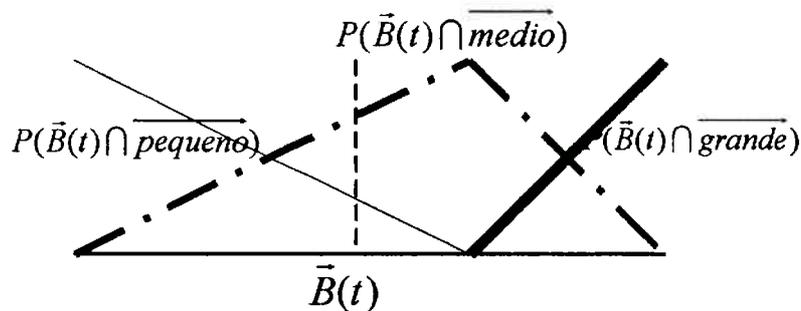
A fim de desenhar a nova função de pertinência, as representações triangulares prévias da Figura 8 são usadas, mas neste caso retirando-se o segmento v de  $t_{1,2}$  e retirando-se o segmento w de  $t_{2,3}$ .

$$\begin{aligned}
 v &= \frac{t_{1-2}(P_{Y_2} - P'_{Y_2})}{P_{Y_2}} \\
 w &= \frac{t_{2-3}(P_{Y_2} - P'_{Y_2})}{P_{Y_2}}
 \end{aligned}
 \tag{71}$$

Desta forma uma faixa vertical é retirada do centro da Figura 7, restando a parte direita e esquerda da figura, que corresponde a Figura 9.d. Estas duas partes laterais restantes da figura são então posicionadas de forma que a distância entre elas seja  $t_{\text{trapézio}}$ . Depois, o espaço entre estas duas partes é obtido como se fosse a parte central da Figura 9.a (direita). Finalmente como é mostrado na Figura 9.e, é necessário que se faça um reajuste do tamanho da Figura 9.d, a fim de ajustar a escala de medidas entre 2 e 12. Isto equivale a alterar-se o valor de  $k$ , o que significa que o plano P está sendo deslocado na direção do vetor perpendicular a este plano.

### ***Funções de pertinência e lei de decisão de Bayes***

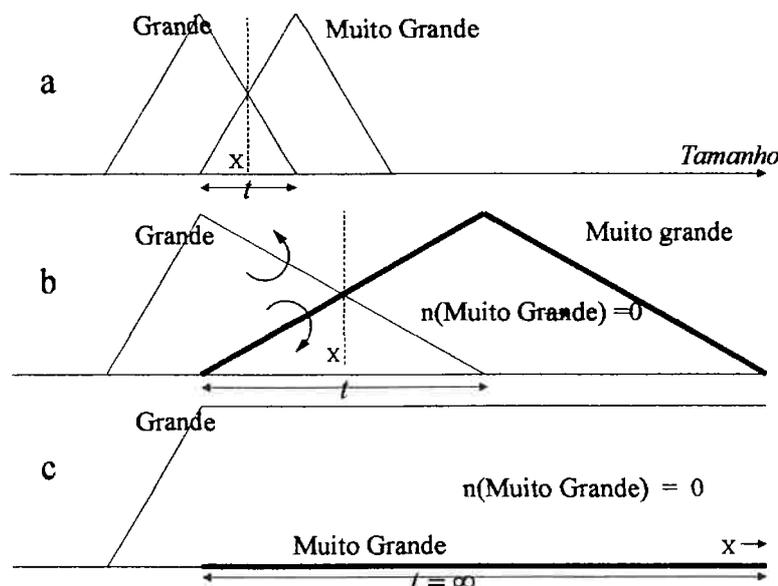
A Figura 6.b pode ser interpretada da forma usual. Contudo levando-se em consideração as probabilidades descritas na parte superior de cada uma delas, pode ser também entendida como um conjunto de funções de decisão de Bayes. Usando o teorema de Bayes [2, 4, 36] nós podemos substituir cada uma destas probabilidades pela probabilidade de um evento  $B(t)$  intersectado com cada uma das categorias consideradas, como é mostrado na Figura 10.



**Figura 10** É possível considerar-se as funções de pertinência como uma função de decisão de Bayes, permitindo a classificação de uma certa medida  $B(t)$  como pertencente à uma classe entre várias classes. No caso desta figura,  $B(t)$  é classificado como "média" porque "média" é a classe com a maior contribuição em  $B(t)$ .

As funções de decisão de Bayes são usadas com a finalidade de se definir em que grupo ou classe, uma determinada informação será classificada. Neste caso, um tipo de decisão "tudo ou nada" é tomada, de forma que a informação é exclusivamente atribuída à uma única classe. ( Este tipo de lógica é usada quando a lógica *fuzzy* realiza agrupamentos no espaço de saída). Em outras palavras, se a probabilidade de um evento pertencer a uma classe for maior que a probabilidade deste pertencer a outra classe, este evento será classificado dentro desta classe de maior probabilidade. Na Figura 11, duas

funções de pertinência, ou seja, duas classes: "grande" e "muito grande", são representadas. A magnitude "x" no eixo horizontal, representa uma fronteira para a qual podemos afirmar com erro mínimo, que o tamanho muda da categoria "grande" para "muito grande".



**Figura 11** (a) O limiar  $x$ , separa os vetores  $B(t)$  que deveriam ser classificados como pertencentes a classe "grande", dos vetores que deveriam ser classificados como muito grandes, de acordo com a lei de decisão de Bayes. (b) Quando a cardinalidade das medidas "muito grande",  $n(\text{muito grande})$  diminui, o limiar  $x$  é deslocado para à direita, significando que um alto valor de  $t$  é necessário para classificar  $B(t)$  como "muito grande". Quando  $n(\text{"muito\_grande"})=0$ , então a distância  $t$  e o limiar  $x$  são infinitos. A função de pertinência etiquetada como "grande" torna-se uma função trapezoidal, enquanto que a função de pertinência "muito grande" é achatada.

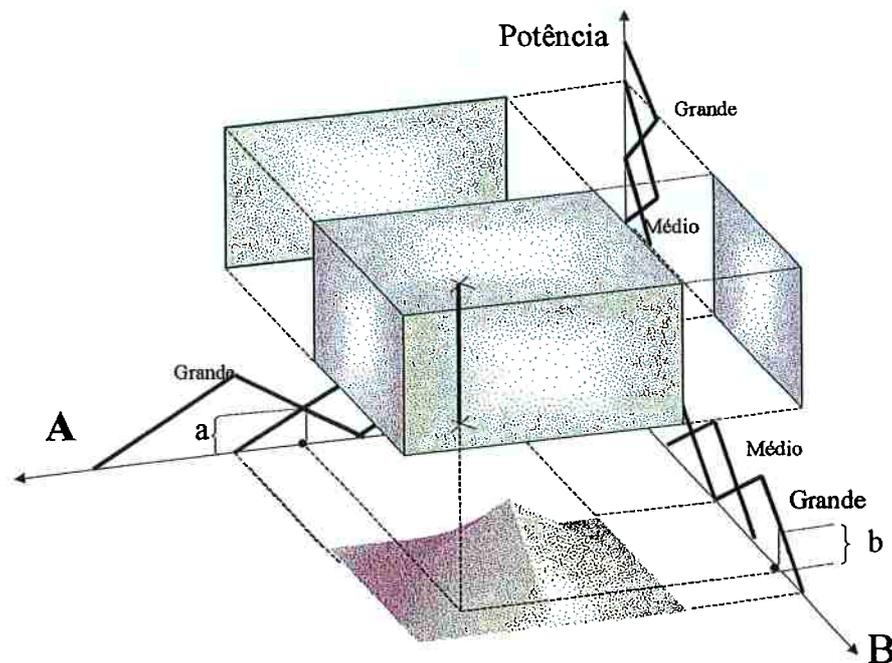
No caso da Figura 11.b, os elementos na classe "muito grande" não são tantos quanto na Figura 11.a. De acordo com as equações anteriores, o intervalo  $t_{3,4}$  torna-se maior, quando a classe "muito grande" tem poucos elementos. Uma das conseqüências deste fato é que a fronteira "x" será deslocada para à direita. Isto significa dizer que se necessita de valores muito altos da magnitude tamanho, para se ter certeza de etiquetar esta magnitude

como "muito grande". O caso extremo, Figura 11.c, é quando o número de elementos na classe "muito grande" é zero. Neste último caso, o lado esquerdo da função de pertinência etiquetada como " muito grande", torna-se zero e assim, o limiar "x" torna-se infinito. Isto significa que se a classe "muito grande" está vazia, não existe certeza alguma para etiquetar um determinado tamanho como pertencente a esta classe. Todas estas considerações são idênticas que aquelas feitas na classificação de padrões através da regra de decisão de Bayes.

**Transformação linear e a regra composicional da inferência.**

**“se ( A é médio e B é grande) então aplica-se uma potência média”**

Este é o tipo de inferência usada na maior parte das vezes na lógica *fuzzy*. Isto implica uma transformação de um espaço bi-dimensional a um espaço uni-dimensional como o representado na Figura 12.

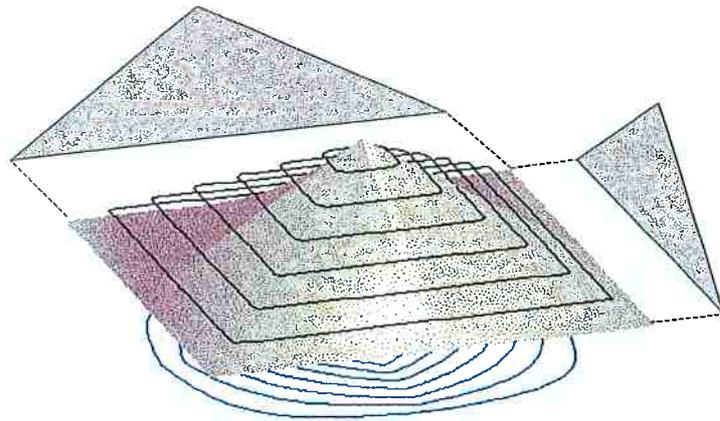


**Figura 12** Representação 3-D da regra: Se A é médio e B é grande, então aplique uma potência média. A pirâmide representa a combinação de funções de pertinência "Médio A", "Grande B" e "Média potência".

A pirâmide na Figura 12 representa a composição de duas funções de pertinência "A média" e "B grande", através da conjunção "e". Um determinado ponto de operação no espaço de tamanhos ( grande, médio,

pequeno), neste caso "(0,a,0) e (0,0,b) " é transformado para produzir o valor da potência. Este valor de saída é um intervalo: média potência, onde o centro de gravidade deste intervalo é normalmente selecionado para fornecer o valor real da potência. A pirâmide representa o resultado da função "e". Este resultado é combinado (normalmente multiplicado) com a função de pertinência da potência correspondente (média potência) resultando o cubo representado na Figura 12. Este cubo é desenhado mais brilhante no centro que nos cantos, porque o valor da composição é maior dentro. Para maior clareza de explicação, estamos representando somente um cubo, ou seja, somente uma regra "se". No caso de mais regras, nós teríamos um conjunto de cubos em forma de cadeia que fariam um mapeamento entre as entradas e as saídas.

Como pode ser observado, a pirâmide na Figura 12 parece ter sofrido uma rotação de 45 graus a partir da posição habitual na qual a função "e" é computada através do cálculo de mínimo. Ao invés deste cálculo habitual, as duas funções de pertinência triangular foram multiplicadas como corresponderia ao cálculo da interseção de dois eventos estatísticos independentes:  $P(A*B)=P(A).(B)$ , gerando a pirâmide com forma mais arredondada da Figura 13.



**Figura 13** A conjunção "e" de duas funções de pertinência triangulares pode ser realizada de duas maneiras: (a) computando-se o mínimo, na qual linhas de nível quadradas são obtidas, ou (b) computando-se a multiplicação de ambas as funções, o que leva a um leve arredondamento da pirâmide. Isto corresponde ao cálculo da interseção de duas funções de pertinência estatisticamente independentes.

As linhas de nível abaixo da pirâmide mostra que a forma da pirâmide é mais arredondada próximo à base e mais quadrada no topo. Isto ocorre porque cada linha de nível está formada por quatro funções hiperbólicas ligadas, do tipo  $z=x.y$ . A linha de nível quadradas, desenhadas acima da pirâmide representa o caso de uma pirâmide gerada mediante o cálculo do mínimo. Com relação a independência de eventos, será repetida aqui a explicação que foi feita na Parte I a fim de derivar algumas regras algébricas, para a regra composicional de inferência e "defuzificação". Foi já explicado que, de acordo com a estatística clássica, um evento A é independente de um evento B se:

$$P(A/B) = P(A) \quad (72)$$

Analisemos esta fórmula, usando nosso conhecimento sobre produtos escalares:

$$P(\vec{A}/\vec{B}) = \frac{n(\vec{A} \cap \vec{B})}{n(\vec{B})} = n(\vec{A} \cap \vec{b}) \quad (73)$$

$$P(\vec{A}) = n(\vec{A} \cap \vec{u}) \quad (74)$$

Se  $\vec{b} = \vec{u}$ , então  $n(\vec{A} \cap \vec{b})$  (o valor da projeção do vetor  $\vec{A}$  sobre  $\vec{b}$ ) iguala a  $n(\vec{A} \cap \vec{u})$  (ou seja o valor da projeção do vetor  $\vec{A}$  sobre  $\vec{u}$ ). Isto significa que, do ponto de vista de  $\vec{A}$ , o universo  $\vec{U}$  e o vetor  $\vec{B}$  são equivalentes ou pelo menos co-lineares (seus módulos podem ser diferentes mas sua direção não). Resumindo a idéia sobre independência podemos formular o seguinte teorema:

*Teorema de independência:*

*Dois eventos  $\vec{A}$  e  $\vec{B}$  são independentes se o espaço amostral de um deles, por exemplo o espaço amostral de um evento  $\vec{A}$  é co-linear com o vetor  $\vec{B}$  ou vice-versa.*

A condição  $n(\vec{A} \cap \vec{b}) = n(\vec{A} \cap \vec{u})$  é como se, de alguma forma, o vetor  $\vec{B}$  estivesse "colorindo" todo espaço do evento  $\vec{A}$ , de forma que o espaço amostral de  $\vec{A}$ , chamemo-lo de  $\vec{U}$ , torna-se um espaço amostral  $\vec{U}$  tipo-B. Da mesma forma, cada um dos eixos do espaço amostral  $\vec{U}$ , é "colorido" pelo evento  $\vec{B}$ , tendo um  $\vec{y}_1$  tipo- $\vec{B}$ , um  $\vec{y}_2$  tipo- $\vec{B}$ , etc. O próximo exemplo esclarecerá algumas dúvidas. Suponhamos que temos duas máquinas que trabalham continuamente e independentemente. Uma destas máquinas produz dois tipos de peças, por exemplo, peças grandes e peças pequenas; enquanto a outra máquina gera cores, por exemplo preto e branco. O output de ambas as máquinas é combinado na saída da máquina de forma que quatro

tipos de combinações são geradas: grande-preto, grande-branco, pequeno-preto, pequeno-branco. Suponhamos que a máquina de cores produz preto 30% do tempo e produz branco 70% das vezes. Quando a máquina de cores está produzindo preto ou branco não se preocupa com o tamanho das peças, de forma que o preto está colorindo 30% das peças, sem importar se são grandes ou pequenas.

$$\begin{aligned} \overrightarrow{\text{Preto}} &= n(\overrightarrow{\text{Preto}} \cap \overrightarrow{\text{grande}})\overrightarrow{\text{Grande}} + n(\overrightarrow{\text{Preto}} \cap \overrightarrow{\text{pequeno}})\overrightarrow{\text{Pequeno}} = \\ &0.3\overrightarrow{\text{Grande}} + 0.3\overrightarrow{\text{Pequeno}} = 0.3(\overrightarrow{\text{Grande}} + \overrightarrow{\text{Pequeno}}) = 0.3\vec{T} \end{aligned} \quad (75)$$

Onde nós chamamos o espaço amostral de tamanhos  $\vec{T}$ , e seu vetor normalizado  $\vec{t}$ . Demonstraremos agora que  $P(\overrightarrow{\text{Grande}}/\overrightarrow{\text{Preto}}) = P(\overrightarrow{\text{Grande}})$  o qual é a condição para a independência dos eventos  $\overrightarrow{\text{Grande}}$  e  $\overrightarrow{\text{Preto}}$ :

$$\begin{aligned} P(\overrightarrow{\text{Grande}}/\overrightarrow{\text{Preto}}) &= n(\overrightarrow{\text{Grande}} \cap \overrightarrow{\text{preto}}) = n(\overrightarrow{\text{Grande}} \cap \frac{0.3\vec{T}}{n(0.3\vec{T})}) = \\ n(\overrightarrow{\text{Big}} \cap \vec{t}.) &= P(\overrightarrow{\text{Big}}) \end{aligned} \quad (76)$$

Neste exemplo o espaço amostral de tamanhos é colorido de duas formas. Isto é, pode ser preto ou branco: 30% dos tamanhos são preto e 70% deles é branco. Estas proporções não variam, apesar de que se considerem peças grandes ou pequenas.

Cada um destes dois espaços amostrais, o espaço amostral de tamanhos,  $\vec{T}$ , e o espaço amostral de cores,  $\vec{C}$ , podem ser considerada sub-espaços amostrais mutuamente independentes.

É possível criarmos um espaço tetra-dimensional a partir dos dois espaços bi-dimensionais. Cada um destes eixos pertencentes a um espaço tetra-

dimensional, é obtido intersectando-se cada eixo de um sub-espço com cada eixo de outro sub-espço, gerando portanto os seguintes eixos:

$$\begin{aligned}
 \vec{y}_{11} &= \overrightarrow{grande} \cap \overrightarrow{preto} \\
 \vec{y}_{12} &= \overrightarrow{grande} \cap \overrightarrow{branco} \\
 \vec{y}_{21} &= \overrightarrow{pequeno} \cap \overrightarrow{preto} \\
 \vec{y}_{22} &= \overrightarrow{pequeno} \cap \overrightarrow{branco}
 \end{aligned} \tag{77}$$

Cada um dos eixos acima são mutuamente ortogonais. Por exemplo, no caso de  $\vec{y}_{11}$  e  $\vec{y}_{22}$  :

$$\begin{aligned}
 n(\vec{y}_{11} \cap \vec{y}_{22}) &= n\left(\left(\overrightarrow{grande} \cap \overrightarrow{preto}\right) \cap \left(\overrightarrow{pequeno} \cap \overrightarrow{branco}\right)\right) = \\
 &= n\left(\left(\overrightarrow{grande} \cap \overrightarrow{pequeno}\right) \cap \left(\overrightarrow{preto} \cap \overrightarrow{branco}\right)\right) = n(\vec{0} \cap \vec{0}) = 0
 \end{aligned} \tag{78}$$

O resto dos casos são calculados de forma análoga :

$$\begin{aligned}
 n(\vec{y}_{11} \cap \vec{y}_{21}) &= n\left(\left(\overrightarrow{grande} \cap \overrightarrow{preto}\right) \cap \left(\overrightarrow{pequeno} \cap \overrightarrow{preto}\right)\right) = \\
 &= n\left(\left(\overrightarrow{grande} \cap 0.3 \cdot \vec{i}\right) \cap \left(\overrightarrow{pequeno} \cap 0.3 \cdot \vec{i}\right)\right) = \\
 &= n\left(0.3 \overrightarrow{grande} \cap 0.3 \overrightarrow{pequeno}\right) = \\
 &0.3 * 0.3 n\left(\overrightarrow{grande} \cap \overrightarrow{pequeno}\right) = 0.3 * 0.3 * 0 = 0
 \end{aligned} \tag{79}$$

Supondo-se que  $P(\overrightarrow{grande}) = 0.2$  e  $P(\overrightarrow{pequeno}) = 0.8$  a probabilidade total do novo espaço amostral,  $\vec{w}$ , gerado pelos quatro eixos é :

$$\begin{aligned}
P(\vec{W}) &= \sum_{i,j=1}^{i,j=2} P(\vec{y}_i) = P(\overline{\text{branco}})P(\overline{\text{grande}}) + P(\overline{\text{branco}})P(\overline{\text{pequeno}}) + \\
&P(\overline{\text{preto}})P(\overline{\text{grande}}) + P(\overline{\text{preto}})P(\overline{\text{pequeno}}) = \\
&0.7 * (0.2 + 0.8) + 0.3 * (0.2 + 0.8) = 0.7 + 0.3 = 1
\end{aligned} \tag{80}$$

O resultado é igual a 1 como acontece com a probabilidade de qualquer espaço amostral.

Nos parágrafos acima, foi mostrado que a operação "e" entre as duas proposições A e B, no antecedente de "Se (A e B) então C" é interpretada como duas operações :

1.Geração de um novo espaço ortogonal  $\vec{u}$ , cuja dimensão é  $\dim(U_A) \times \dim(U_B)$ .

2.O valor de cada componente de certo vetor  $V(t)$ ,  $n(\vec{V}(t) \cap \vec{y}_{ij})$  no novo espaço U é:  $n(\vec{V}(t) \cap \vec{y}_{ij}) = n(\vec{V}(t) \cap \vec{y}_i) * n(\vec{V}(t) \cap \vec{y}_j)$  onde cada um dos  $\vec{y}_i$  são os eixos orto-normais do sub-espaço  $U_A$ , e cada um dos  $\vec{y}_j$  são os eixos orto-normais do sub-espaço  $U_B$ . Esta multiplicação é equivalente a multiplicar-se as probabilidades de dois eventos independentes, fornecendo a probabilidade da interseção de ambos.

Finalmente, C é obtido através de uma transformação linear que será explicada usando-se o mesmo exemplo acima.

Tomemos as seguintes regras :

- ◆ *Se ( a peça é preta e grande ), coloque-a ao lado direito.*
- ◆ *Se ( a peça é branca e pequena), coloque-a ao lado esquerdo.*
- ◆ *Se ( a peça é preta e pequena) coloque-a ao lado direito.*

É possível expressar-se as regras acima como uma transformação linear de

um espaço de entrada tamanho-cor, em um espaço de posições como se segue:

$$\begin{aligned} \begin{bmatrix} \overrightarrow{n(V(t) \cap \overrightarrow{right})} \\ \overrightarrow{n(V(t) \cap \overrightarrow{left})} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \overrightarrow{n(V(t) \cap \tilde{y}_{11})} \\ \overrightarrow{n(V(t) \cap \tilde{y}_{12})} \\ \overrightarrow{n(V(t) \cap \tilde{y}_{21})} \\ \overrightarrow{n(V(t) \cap \tilde{y}_{22})} \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \overrightarrow{n(V(t) \cap \tilde{y}_{i=1})} \cdot \overrightarrow{n(V(t) \cap \tilde{y}_{j=1})} \\ \overrightarrow{n(V(t) \cap \tilde{y}_{i=1})} \cdot \overrightarrow{n(V(t) \cap \tilde{y}_{j=2})} \\ \overrightarrow{n(V(t) \cap \tilde{y}_{i=2})} \cdot \overrightarrow{n(V(t) \cap \tilde{y}_{j=1})} \\ \overrightarrow{n(V(t) \cap \tilde{y}_{i=2})} \cdot \overrightarrow{n(V(t) \cap \tilde{y}_{j=2})} \end{bmatrix} \end{aligned} \quad (81)$$

Desta forma o vetor  $\overrightarrow{V(t)}$  pode ser expresso de duas maneiras, dependendo do espaço sobre o qual  $\overrightarrow{V(t)}$  é projetado.  $\overrightarrow{V(t)}$  é expresso como vetor de quatro componentes no espaço de entrada, enquanto que no espaço de saída,  $\overrightarrow{V(t)}$  é expresso como um vetor bi-dimensional. Os elementos de cada coluna da matriz de transformação 2x4 são as projeções sobre os dois eixos do espaço de saída de cada um dos quatro eixos do espaço de entrada

### ***Conclusões da Parte II***

O presente trabalho propõe que os eventos  $fuzzy$  são vetores em um espaço Euclidiano que é definido usando-se um tipo específico de produto escalar  $n(\vec{A} \cap \vec{B})$ , sendo  $\vec{A}$  e  $\vec{B}$  dois eventos  $fuzzy$ . A probabilidade de um evento é um tipo especial de medida que é definida como o módulo da projeção de um evento  $fuzzy$  sobre um vetor unitário específico que é a soma

normalizada dos vetores da sua base de eixos. Este vetor unitário é o vetor característico do plano de iso-probabilidade P. Em outras palavras, todos os vetores da lógica *fuzzy* com suas origens no centro das coordenadas e as pontas das flechas neste plano, têm a mesma probabilidade. Por conseguinte, qualquer tipo de evento, dentro de qualquer trajetória pertencente ao plano P, tem a mesma probabilidade.

Qualquer mapeamento de entrada-saída, com um repertório limitado de pares de entrada-saída, requer um desenho correto do espaço de entrada e do espaço de saída. Para este propósito, o princípio de máxima entropia afirma que a melhor distribuição de probabilidades para as categorias do espaço de entrada ou saída, é a aquela que maximiza a entropia em qualquer um destes espaços, contanto que satisfaça as informações ou restrições do problema. Se a única informação que se tem, é a que existem n eventos mutuamente exclusivos, então a probabilidade de qualquer um deles é a mesma:  $1/n$ . No caso do mapeamento de entrada-saída, no qual a única informação que temos sobre o espaço de entrada ou de saída é a proporção relativa das categorias envolvidas, o plano de iso-probabilidade fornece a melhor aproximação para representar o espaço de entrada ou saída.

Várias trajetórias podem ser representadas neste plano como a que gera as funções de pertinência triangular ou trapezoidal amplamente utilizadas. Todos os pontos, ou seja, eventos-vetores, em qualquer destas trajetórias, têm a mesma probabilidade, obtida com uma combinação de probabilidades das diferentes categorias para cada evento. Estas propriedades fundamentam um método amplamente usado para desenhar funções de pertinência. Este

método consiste em fazer constante a soma das funções de pertinência ao longo do domínio de entrada ou saída das variáveis, o qual é consistente com as demonstrações feitas neste artigo. No caso de se ter um conhecimento mais aprofundado das características dos espaços de entrada ou saída, um cálculo mais analítico para a maximização da entropia do sistema é possível, como aquele que usa a técnica dos multiplicadores de Lagrange. Contudo, a complexidade desta abordagem, é uma boa razão para continuar usando-se métodos tradicionais como o método  $\alpha$ -cuts para o desenho das funções de pertinência.

Finalmente a regra composicional de inferência é interpretada como uma transformação linear, na qual o espaço de entrada é obtido através do produto cartesiano das categorias das variáveis de entrada. Algumas das demonstrações que permitem derivar lógica *fuzzy* e a própria estatística a partir da álgebra euclidiana, são também usadas para a implementação de um novo paradigma de rede neural [24, 25, 28], que será mostrada na próxima parte da tese (Parte III).

### **PARTE III APLICAÇÃO DA ESTATÍSTICA EUCLIDIANA NO DESENVOLVIMENTO DE UMA NOVA REDE NEURAL (REDE TALÂMICA)**

A matemática euclidiana apresentada na primeira parte foi utilizada para fundamentar teoricamente uma sub-área da inteligência artificial, a lógica 'fuzzy'. Utilizaremos os mesmos fundamentos matemáticos para aprofundarmos em outra sub-área da mesma inteligência artificial, as redes neurais. Desta maneira pode-se constatar que as duas disciplinas podem ter fundamentos matemáticos comuns. Será feita uma abordagem que consiste em desenvolver um novo modelo de rede neural com importantes semelhanças com uma estrutura que realmente existe no cérebro, o tálamo. Embora os fundamentos biológicos no qual está baseado este modelo sejam de grande importância, será dada mais atenção à matemática subjacente ao modelo. Utilizando-se esta matemática mostrar-se-á que a arquitetura desta rede neural nos permite criar uma base de eixos para redução da dimensão dos padrões que são apresentados à rede. Desta maneira se conseguirá uma importante compressão da informação sensorial que flui através do tálamo em direção a camadas superiores no cérebro. Compressão e redução da dimensão dos dados, são técnicas muito utilizadas na prática para otimizar o armazenamento e facilitar a comunicação de dados. Mostrar-se-á como a rede efetua os dois tipos de algoritmos de Gram-Schmidt, já apresentados na primeira parte, para a criação de uma base de eixos ortogonais. No primeiro algoritmo de Gram-Schmidt a posição dos eixos varia dependendo da

seqüência de apresentação dos padrões, enquanto que o segundo algoritmo cria uns eixos cujas posições são fixas e coincidem com os chamados componentes principais do conjunto de padrões.

Começaremos com uma revisão histórica visando enquadrar a presente pesquisa dentro da área das redes neurais artificiais. Em seguida serão apresentadas as características essenciais do tálamo para depois, explicarmos os algoritmos de Gram-Schmidt no contexto desta rede.

### ***Revisão histórica.***

Embora a área das redes neurais seja uma das áreas de pesquisa mais fecundas na atualidade, poucas redes neurais têm sido desenvolvidas tentando modelar estruturas cerebrais. A maioria das tentativas provenientes da área da matemática e de engenharia conformam-se com pegar algumas idéias da neurofisiologia e logo continuar o caminho, com o único critério de aplicabilidade da rede. Os cientistas da área biológica modelam computacionalmente algumas estruturas cerebrais sob o ponto de vista da dinâmica das mesmas sem considerar nenhuma capacidade de aprendizado. Estes modelos computacionais na área biológica, analisam as oscilações [37, 39, 43] em grupos de neurônios, as alternâncias em atividade entre áreas cerebrais, etc. para dar solução a vários problemas médicos como a epilepsia. Do ponto de vista dinâmico foram realizadas algumas simulações do tálamo [19, 39] embora o aspecto de aprendizado não tenha sido ainda considerado.

Entre as tentativas de modelar o cérebro na área da engenharia, uma das redes mais conhecidas é a Self Organizing Map (SOM) de Teuvo Kohonen [12], que tenta simular o córtex cerebral usando uma rede neural competitiva constituída por duas camadas. Entre as características mais significativas da rede SOM, está o fato de conservar a topologia dos dados externos na representação interna da rede neural. Este fato não contradiz as pesquisas sobre o cérebro, as quais demonstram a existência, no córtex cerebral, de mapas somato-tópicos nos quais estão representadas todas as áreas do corpo humano. Um dado que não é tão conhecido, é que no tálamo<sup>1</sup> também existem este tipo de mapas nos quais a topologia dos dados externos é

---

<sup>1</sup> O tálamo pode ser considerado a etapa anterior de processamento ao córtex.

conservada na representação interna. Além disto a arquitetura da rede de Kohonen, segundo recentes dados neurofisiológicos, parece assemelhar-se mais ao tálamo que ao córtex cerebral o qual motivou sua criação. A estrutura em dupla camada da rede SOM, com uma das camadas competitiva, é mais similar a estrutura do tálamo, visto que o córtex cerebral tem uma estrutura em multi-camadas onde os neurônios competitivos propriamente ditos (inibitórios) misturam-se com os neurônios de projeção (excitatórios) à outras camadas. Além das semelhanças existem importantes diferenças entre a rede SOM e o modelo do tálamo, embora neste item ressaltemos somente as analogias. O fato de ressaltarmos algumas características comuns entre a rede SOM e a rede talâmica facilita o entendimento da rede talâmica para as pessoas que já conhecem a rede SOM. É por esse motivo que vamos apresentar agora, brevemente, algumas características comuns da SOM e da rede talâmica:

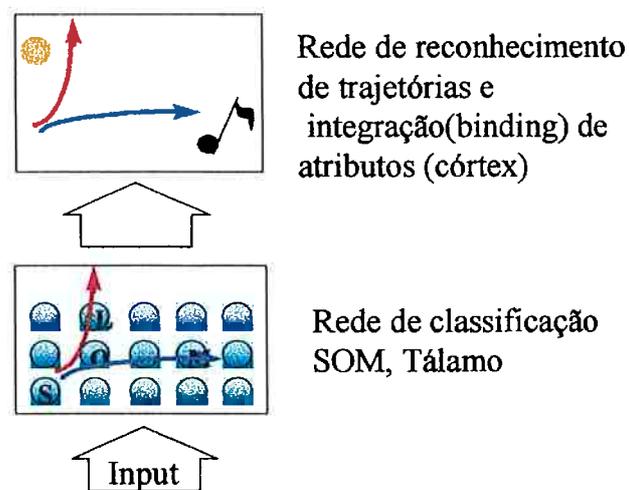
- Estrutura de dupla camada. Ambas as redes, a rede de Kohonen e a rede talâmica possuem uma camada de entrada e uma camada competitiva.

- O desempenho da camada competitiva nas duas redes é melhor com uma disposição esférica (como aquela da rede SS-SOM) onde a distorção das bordas é eliminada. O tálamo funciona como uma rede deste tipo.

- Na rede SOM os neurônios da camada reticular pode ter “consciência”, um eufemismo, para dizer que, depois da vitória de um neurônio na camada competitiva este neurônio cede a possibilidade de vencer a outro neurônio da

mesma camada. Nos neurônios da camada competitiva do tálamo acontece o mesmo fenômeno.

Normalmente cada tipo de padrão apresentado à rede, ativa um neurônio diferente na camada competitiva. A conservação da topologia na representação interna implica que tipos de padrões parecidos produzem a ativação de neurônios reticulares próximos entre si na camada reticular.



**Figura 14** Duas redes neurais encadeadas para classificação e integração.

Esta propriedade vem sendo utilizada para o reconhecimento de voz [13, 20, 21]. Neste caso cada neurônio competitivo representa um determinado “fonema”<sup>2</sup> do mapa fonotópico, o qual é constituído pela camada inteira de neurônios competitivos. A trajetória, resultante da união em seqüência dos neurônios vencedores durante a apresentação de uma determinada palavra, poderia servir de entrada para outra rede permitindo o reconhecimento de palavras. Desta maneira, diferentes trajetórias representam diferentes palavras.

<sup>2</sup> Utilizamos este conceito para maior simplicidade embora não seja totalmente apropriado neste caso.

Observe-se que a seqüência temporal de fonemas (padrões) define a seqüência dos neurônios na camada competitiva. Se a palavra é formada por um fonema isolado (um padrão), um só neurônio é ativado na camada competitiva.

A rede talâmica também produz trajetórias. A diferença entre a rede talâmica e a rede SOM, é que na rede talâmica, um padrão isolado (um fonema, um desenho etc. ) pode gerar toda uma trajetória na camada competitiva, trajetória esta composta não por um só neurônio mas por uma seqüência destes neurônios. Por este motivo a correspondência de um padrão (um fonema por exemplo ) com um neurônio não existe na rede talâmica. Na rede talâmica, um padrão isolado pode gerar uma seqüência de ativações de neurônios na camada competitiva, sendo cada neurônio, o responsável por um atributo do padrão de entrada. Por exemplo, se o padrão de entrada for um desenho com vários atributos geraria-se na camada competitiva do tálamo, uma seqüência de ativação de vários neurônios. Esta seqüência de atributos, veremos, equivale à seqüência dos componentes principais de cada padrão. Se fosse utilizada a rede de Kohonen a camada competitiva forneceria um só neurônio em resposta a cada padrão.

A trajetória seqüencial dos neurônios define portanto, a identidade dos padrões, mas para isto é necessário uma outra rede neural que tenha como padrão de entrada a seqüência de neurônios da camada competitiva, e como padrão de saída um só neurônio representando o padrão. Isto poderia ser feito superpondo-se à rede talâmica um modelo de rede neural que simule a atividade de processamento e integração temporal de padrões no córtex cerebral. Embora tenhamos como hipótese de trabalho para o desenho do

modelo do tálamo a existência de uma outra rede cortical para integração temporal dos atributos gerados pela rede talâmica, esta tese basear-se-á somente no modelo talâmico. O hipótese da integração temporal de atributos no córtex [14, 41, 43] foi apresentado para enquadrar a pesquisa num contexto mais abrangente. Neste módulo cortical, os diferentes atributos de um padrão seriam novamente integrados em um só neurônio que representa o padrão de entrada. Se por exemplo a rede talâmica e a rede cortical fossem aplicadas ao reconhecimento de voz, um “fonema” reconhecido através de um só neurônio na SOM, provocaria a ativação de vários neurônios, seqüencialmente, no módulo talâmico. Esta seqüência de neurônios integrariam-se posteriormente em um só neurônio no módulo cortical. Deste modo cada fonema geraria a atividade de um neurônio cortical. A seqüência de ativação dos neurônios do módulo cortical representaria uma palavra. Estes neurônios corticais seriam especializados em processar seqüências formando assembléias. Portanto, uma determinada assembléia de neurônios corticais, seria responsável pelo processamento do conteúdo de uma palavra. É possível que esta apresentação comparativa, longe de esclarecer, tenha gerado muitas questionamentos sob a rede talâmica, especialmente na situação de não estar familiarizado com a rede SOM. Nas seções seguintes desenvolveremos a matemática que fornece o embasamento para o entendimento de características da rede talâmica. Nas seguintes seções serão apresentadas algumas características neuro-fisiológicas do tálamo e dos neurônios desta estrutura cerebral. Esta apresentação e a posterior modelagem do tálamo será, de forma inevitável, muito simples, visando utilizar as características do tálamo que permitem que este faça uma extração

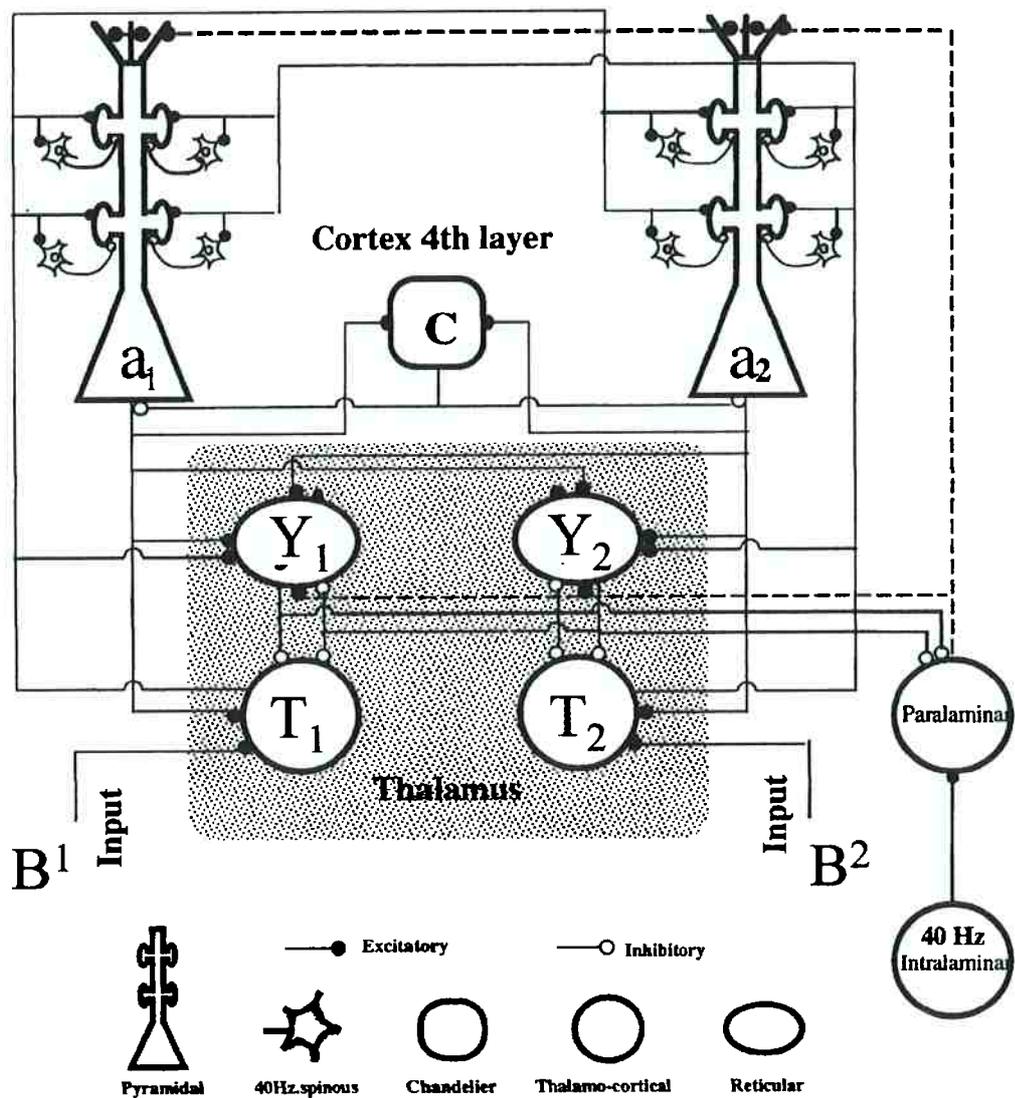
de eixos ortogonais que codificam os componentes principais dos padrões sensoriais. O nosso objetivo neste trabalho, é criar uma rede neural para o processamento de padrões, aplicando-se a matemática apresentada na parte I da tese. Casualmente, as equações de ortogonalização apresentadas nessa parte I podem ser implementadas na arquitetura de conexões entre neurônios do tálamo. Algumas propriedades destes neurônios, que serão mostradas a seguir também parecem favorecer a hipótese de uma extração de componentes principais no tálamo. A fim de dar ao trabalho uma orientação eminentemente prática, somente serão utilizadas as características do tálamo e de seus neurônios, as quais permitem esta extração de componentes principais. Serão ignoradas portanto outras características neurofisiológicas importantes, mas irrelevantes sob a perspectiva da engenharia. Contudo se estas características que omitiremos fossem consideradas, não haveria prejuízo algum na função de extração de atributos da rede talâmica. Por exemplo os neurônios são considerados essencialmente binários ignorando-se a dinâmica intrínseca do potencial de ação (disparo do neurônio) que foi descrita magistralmente pelos cientistas Hodgkin e Huxley [10]. Outra característica omitida é que os neurônios do tálamo apresentam, além do tipo comum de disparo, um outro tipo de disparo em alta frequência (200-300 Hz) denominado, na língua inglesa, disparo do tipo 'burst' ou rajada. Esta característica seria muito importante na hora do tálamo comunicar o resultado dos seus cálculos ao córtex. Embora o aluno tenha estudado teoricamente esta característica em outros trabalhos [26, 27, 28, 29] o disparo em alta frequência não será incorporado no nosso modelo de rede neural pois a interação tálamo-cortical esta fora do escopo desta tese de doutorado.

### ***Localização e estrutura do tálamo***

O tálamo [16, 17, 19, 22, 23] está formado por duas camadas, uma camada interna, de entrada formada por neurônios tálamo-corticais  $T_i$  que são excitatórios (as sinapses excitatórias estão indicadas por um ponto preto na Figura 15), e uma outra camada externa, denominada reticular, de neurônios inibitórios  $Y_i$  (a coincidência em notação com os eixos  $\bar{Y}_i$  é intencional, e as sinapses inibitórias estão indicadas por um ponto branco na figura).

Os neurônios tálamo-corticais da primeira camada (camada interna do tálamo), recebem a informação dos padrões da entrada  $B_j$ , onde cada entrada esta codificada como  $B_i^j$ .

O índice  $j$  acima, indica a entrada, e o índice  $i$  abaixo, indica o número do padrão. Estes neurônios tálamo-corticais projetam-se sobre os neurônios reticulares (segunda camada do tálamo) e sobre os neurônios da quarta camada do córtex.



**Figura 15** Estrutura e conexões do tálamo e da camada IV do córtex. As sinapses excitatórias estão indicadas por um ponto preto enquanto que as inibitórias estão indicadas por um ponto branco. Os cruzamentos entre conexões estão indicados por arcos. Os cruzamentos sem arcos são bifurcações.

Os neurônios reticulares da segunda camada do tálamo (camada mais externa) são neurônios inibitórios os quais enviam projeções intra-talâmicas para outros neurônios reticulares da mesma camada (projeções laterais) e para neurônios tálamo-corticais (da camada interna do tálamo) (vide Figura 15). Os neurônios intralaminares mostrados na figura, são neurônios que disparam numa frequência de 40 Hz [38]. Estes neurônios junto com os neurônios para-laminares impõem uma certa ritmicidade ao sistema, permitindo sincronizar a parte talâmica com a parte cortical [18]. Neste trabalho, somente o tálamo propriamente dito vai ser considerado. Por este motivo estes neurônios intra e para-laminares, assim como todos os neurônios corticais não vão ser considerados.

### ***Modelagem dos pesos sinápticos***

A comunicação entre os neurônios realiza-se através das sinapses. O valor da eficiência desta comunicação se denomina, dentro da área das redes neurais artificiais, peso sináptico, e é representado pela letra  $w_{ij}$  (do inglês weight), onde o primeiro índice (aqui é o  $i$ ) corresponde ao neurônio post-sináptico enquanto o segundo (neste caso o  $j$ ) corresponde ao neurônio pre-sináptico. Estes pesos sinápticos [1, 9, 30] serão modelados, neste trabalho da seguinte forma: Sendo  $T_j$  o neurônio pré-sináptico e  $Y_i$  o neurônio pós-sináptico, o peso entre eles é  $w_{ij} = P(Y_i/T_j)$ . Normalmente, tanto os neurônios inibitórios quanto os neurônios reticulares não têm pesos sinápticos modificáveis nas suas projeções sobre outros tipos de neurônios. É por este motivo que o mecanismo de inibição lateral (competição) entre neurônios reticulares não utiliza pesos modificáveis (Os pesos podem ser considerados iguais a 1) Contudo as projeções de volta dos neurônios reticulares para os tálamo-corticais requerem pesos modificáveis para reproduzir as equações de Gram-Schmidt. Este dilema pode ser assim expresso: Como podem neurônios sem pesos modificáveis comportar-se em alguns locais como se tivessem? A resposta é, mediante uma complexa estrutura chamada glomérulo em cuja arquitetura intervem um outro neurônio que ainda não foi introduzido: o neurônio de circuito local. Propõe-se que os glomérulos permitem que os pesos sinápticos da conexão que vai dos neurônios reticulares aos neurônios tálamo-corticais sejam ajustados mediante a seguinte expressão  $P(B_i^j/Y_i)$ . Contudo este tema não será discutido nesta tese por sair

fora do escopo da mesma. O leitor pode encontrar maior informação na referência [28]. Este resultado difere do resultado caso o cálculo fosse baseado em  $P(T^j/Y_i)$  (aplicando-se a regra de reforço sináptico explicada no início deste segundo item e observando a Figura 15). A justificativa biológica da afirmação anterior não será abordada aqui, pela orientação prática deste trabalho. O leitor interessado pode encontrar esta justificativa na referência [28].

### ***Propriedades dos neurônios da camada reticular***

A probabilidade de um disparo de um neurônio reticular, calcula-se tendo em vista várias propriedades importantes destes neurônios.

a/ Propriedade denominada homeostasis sináptica que permite normalizar os pesos ou eficiências sinápticas que chegam a cada neurônio reticular. Deste modo será possível conseguir que a atividade de cada neurônio reticular se adapte ao número de entradas sendo esta atividade máxima quando o número máximo de entradas ao neurônio estejam sendo ativadas. Posteriormente será mostrado que os pesos a cada neurônio reticular representam as coordenadas de cada componente principal. Os componentes principais serão normalizadas por causa desta propriedade.

b) Os neurônios reticulares possuem uma atividade oscilatória intrínseca que propomos que sirva para o cálculo, em paralelo, do neurônio com uma maior ativação dentre o conjunto de neurônios da camada reticular. O valor da ativação provinda das entradas sinápticas seria superposta a esta oscilação, sendo o primeiro neurônio que dispare. aquele neurônio cuja atividade alcançara um determinado limiar. Este cálculo evitaria um processamento em série entre todos os neurônios da camada reticular para calcular o máximo, o qual precisaria muito tempo para ser realizado.

Dentro desta filosofia de obtenção do máximo, poderíamos propor uma solução mais elegante tendo em vista que os neurônios reticulares são especialmente sensíveis a incrementos na própria ativação. Assim o limiar de

disparo não seria um valor absoluto de ativação e sim um valor de incremento da ativação. Quando o incremento da ativação supere este limiar, o neurônio de maior ativação será o primeiro em disparar. Com esta solução a atividade oscilatória intrínseca dos neurônios reticulares deveria multiplicar (não ser somada) aos “inputs”. Esta solução evita esperar que um determinado neurônio atinja um determinado teto para começar a disparar, calculando-se o máximo já no início da oscilação. Para uma maior eficiência neste tipo de computação, a oscilação deveria possuir uma forma de sigmoide no tramo ascendente o qual parece confirmado nos neurônios reticulares reais.

### ***Propriedades dos neurônios da camada tálamo-cortical.***

Referindo-se aos neurônios tálamo-corticais (ou da primeira camada no tálamo) possuem largos períodos de hiperpolarização <sup>3</sup> [16, 22] depois dos seus disparos. Esta propriedade especial dos neurônios tálamo-corticais permite a soma das inibições produzidas pelos neurônios reticulares nos sucessivos instantes em que os neurônios reticulares disparam.

---

<sup>3</sup> Hiperpolarização significa que o potencial da membrana do neurônio fica mais negativo que o potencial de repouso normal. Por exemplo se o potencial de membrana em repouso é de -55mV hiperpolarizando poderia-se ter um potencial de -65mV

### *Computação de Gram-Schmidt no modelo do tálamo.*

A seguir explicar-se-á como os neurônios tálamo-corticais computam, individualmente, o teorema de Gram-Schmidt. Depois, este teorema, será demonstrado sobre um padrão mais complexo constituído por um conjunto de entradas individuais. Como foi antecipado, na introdução desta terceira parte, a computação tradicional de Gram-Schmidt pode demonstrar-se no tálamo, quando os padrões são apresentados numa seqüência específica, um de cada vez, e além disto, cada padrão permanece na entrada até que os pesos sinápticos estejam estabilizados. Voltemos brevemente a Parte I da tese para desenvolver, desde aquelas bases matemáticas o resto da explicação. Os dois membros da igualdade da equação 7, na Parte I, poderia ser interpretada como dos eventos equilibrando-se mutuamente. O evento  $\bar{B}$  no membro da esquerda da igualdade poderia ser uma entrada excitatória a um neurônio tálamo-cortical enquanto a somatória do lado direito da igualdade poderia ser a soma total das inibições desde os neurônios reticulares  $\vec{Y}_i$  sendo ponderadas pelos pesos  $P(\bar{B}/\vec{Y}_i)$ . Este equilíbrio de forças entre excitação e inibição produziria, neste caso, que o evento ‘disparo do neurônio tálamo-cortical’  $\vec{T}^j$  seja o evento nulo.

$$\vec{T}^j = \sum \text{inputs} = \bar{B}^j - \sum_{i=1}^n P(\bar{B}^j / \vec{Y}_i) \cdot \vec{Y}_i = 0 \quad (82)$$

Se um neurônio reticular fosse desconectado a ativação talâmica seria exatamente igual a esta contribuição que não é fornecida.

$$\vec{T}_n = P(\vec{B} / \vec{Y}_n) \cdot \vec{Y}_n = \vec{B}^j - \sum_{i=1}^{n-1} P(\vec{B}^j / \vec{Y}_i) \cdot \vec{Y}_i \quad (83)$$

Finalmente se o atributo  $\vec{Y}_n$  representa uma característica exclusiva da entrada  $\vec{B}^j$  então  $P(\vec{B}^j / \vec{Y}_n) \rightarrow 1$ , e portanto obteremos a iteração enésima de Gram-Schmidt:

$$\vec{T}_n = \vec{Y}_n = \vec{B}^j - \sum_{i=1}^{n-1} P(\vec{B}^j / \vec{Y}_i) \cdot \vec{Y}_i \quad (84)$$

Do mesmo modo todas as iterações anteriores de Gram-Schmidt permitem obter uma característica exclusiva do padrão atual que não estava presente nos padrões previamente apresentados. Até agora temos estudado o teorema de Gram-Schmidt aplicado a neurônios tálamo-corticais individuais mas também pode ser aplicado a um padrão complexo constituído pelo conjunto (soma) de todas as entradas individuais  $\vec{B}^j$ . Por causa da linearidade de todas as operações que aparece no cálculo de Gram-Schmidt:

$$\vec{Y}_k = G_{i=k} \left( \sum_{j=1}^m \{ \vec{B}^j \} \right) = \sum_{j=1}^m \{ G_{i=k} ( \vec{B}^j ) \} \quad (85)$$

Portanto o resultado da aplicação de Gram-Schmidt a um padrão composto pela soma de várias entradas, é equivalente a calcular-se a soma dos resultados de Gram-Schmidt aplicado à cada uma destas entradas.

A competição entre os neurônios [39] dentro da camada reticular pode ser entendido como um processo de busca da maior das funções de decisão bayesianas [2,4]  $P(\vec{T} \cap \vec{Y}_k)$  nas quais o  $\vec{T}$  é  $\vec{T} = \sum_{j=1}^m \{ \vec{T}^j \}$ , e  $\vec{Y}_k$  representa a ativação de cada um dos neurônios reticulares candidatos.

Neste caso a regra de decisão de Bayes pode ser escrita como segue:

$$\vec{Y}_k = \vec{Y}_k^* / \forall \vec{Y}_k \quad P(\vec{T} \cap \vec{Y}_k^*) \geq P(\vec{T} \cap \vec{Y}_k) \quad (86)$$

A equação anterior representa o processo competitivo que é desenvolvido

na camada reticular do tálamo. Significa que, cada vez que um padrão é apresentado, o neurônio reticular mais ativo é posto em destaque. Os neurônios reticulares mais ativos ao longo das sucessivas apresentações de padrões ao sistema, constituem uma base de eixos ortogonais já que.

$$n(\vec{Y}_i \cap \vec{Y}_j) = 0 \quad \forall i \neq j \quad (87)$$

Uma vez obtidos todos os eixos, qualquer vetor  $\vec{B} = \sum_{j=1}^m \vec{B}^j$  pode ser representado sobre os eixos:

$$\vec{B} = \sum_{j=1}^m \vec{B}^j = \sum_{j=1}^m \left[ \sum_{i=1}^n P(\vec{B}^j / \vec{Y}_i) \vec{Y}_i \right] \quad (88)$$

Tendo em vista que  $\vec{Y}_i$  representa a ativação dos neurônios reticulares que produzem inibição sobre os neurônios tálamo-corticais, e que  $P(\vec{B}^j / \vec{Y}_i)$  é o peso que modula esta inibição de  $Y_i$  em direção a  $T^i$ , a equação 88 representa o padrão das inibições desde os neurônios reticulares que, no fim do processo de extração de atributos, contrapesa (anulando) o padrão  $\vec{B}$  nos neurônios tálamo-corticais. Desta maneira as inibições formam uma imagem de espelho das excitações sobre os neurônios tálamo-corticais. Quando o padrão de excitações está incompleto, a imagem de espelho produzida pelas inibições continua incluso em neurônios onde a excitação já não existe. Isto permite completar o padrão das excitações.

No seguinte exemplo realizar-se-á um análise da computação no tálamo quando dois padrões  $\vec{b}_1$  e  $\vec{b}_2$  são apresentados em seqüência. Pelo fato deste exemplo e da Figura 16 correspondente terem sido retiradas de um artigo publicado anteriormente pelo aluno [28], a notação muda um pouco: algumas letras maiúsculas são minúsculas no exemplo e os sub-índices e super-índices

são um pouco diferentes dos usados anteriormente. Durante a apresentação do primeiro padrão  $\bar{b}_1$  (Figura 16.a.1 e Figura 16.a.2) os três primeiros neurônios tálamo-corticais  $t_1$ ,  $t_2$  e  $t_3$  ativam-se com uma determinada probabilidade que é a mesma nos três, pelo fato do padrão  $\bar{b}_1$  ser igual nas três entradas. A apresentação do segundo padrão (Figura 16. b.1) corresponde ao padrão  $b_2$  que, neste exemplo, está caracterizada pela ativação dos neurônios talâmicos  $t_2$ ,  $t_3$ ,  $t_4$  e  $t_5$ . As conexões inibitórias que partem dos neurônios reticulares, não são modificáveis mas, por causa da existência de um tipo de neurônio chamado de circuito-local, poderiam mudar quando fazem sinapse com os neurônios tálamo-corticais, como foi já mencionado. Também, como foi explicado antes, o peso destas sinapses poderia ser calculado como a probabilidade da entrada  $b^j$  condicionado pelo disparo de um neurônio reticular  $y$ ,  $P(b^j/y)$ , de forma que a probabilidade da inibição sobre o neurônio tálamo-cortical seja  $P(b/y)P(y)$ .

Quando o primeiro padrão  $b_1$  é apresentado, (Figura 16.a.1), um determinado neurônio reticular, por exemplo o neurônio  $y_1$ , na camada competitiva de neurônios reticulares ganha a competição para representar o padrão de entrada. Durante esta apresentação, os pesos das sinapses entre cada neurônio tálamo-cortical  $t_j$  e o neurônio reticular  $y_1$ ,  $P(y_1/t_j)$  são atualizados. Após este reforço, cada vez que  $\bar{b}_1$  é apresentado, este gerará a ativação de  $y_1$ , de forma que  $P(y_1)=P(b_1)$ . Esta igualdade representa a primeira iteração do algoritmo de Gram-Schmidt.

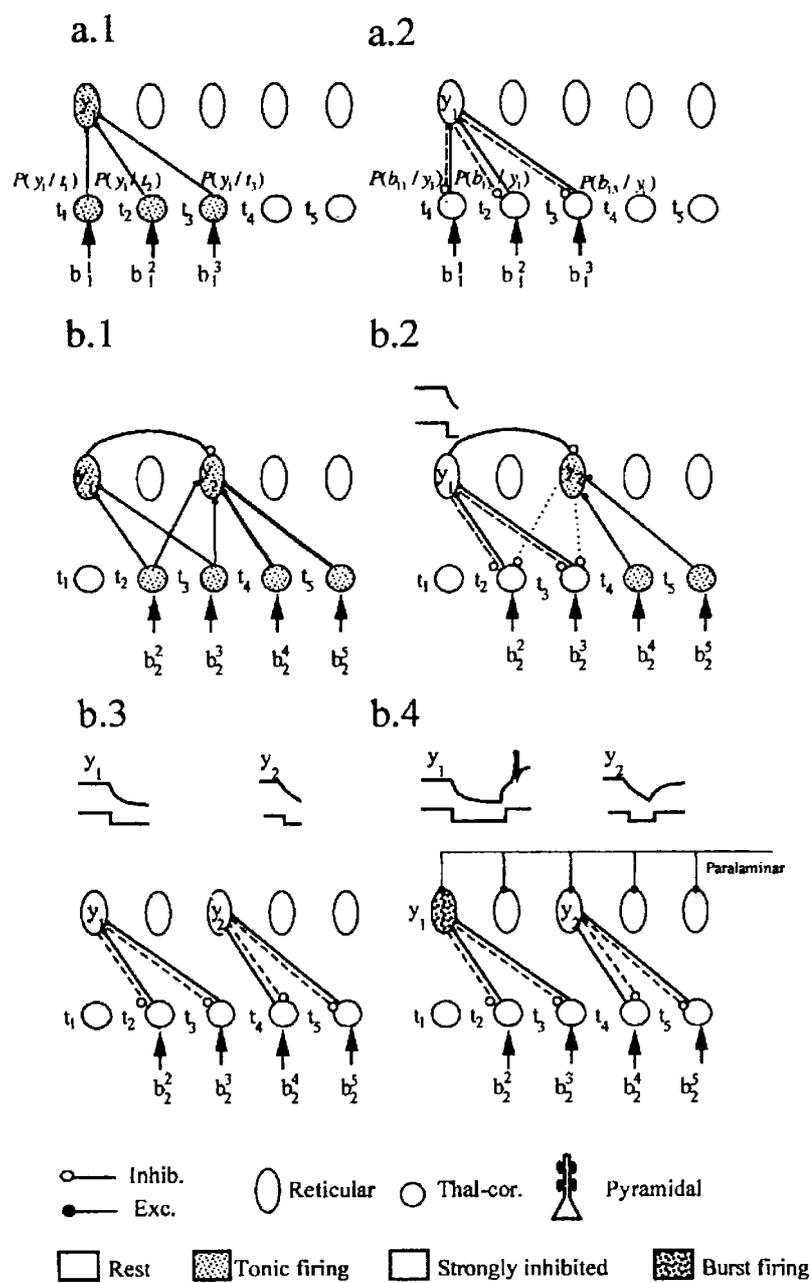


Figura 16

Os pesos inibitórios das projeções de volta  $P(\vec{b}_n^j / \vec{y}_i)$  desde os neurônios reticulares também são reforçados (sendo  $j$  cada uma das unidades de entrada do padrão  $\vec{b}_n$ ). Por causa do incremento de inibição subsequente ao reforço os neurônios talâmicos são finalmente silenciados (ver Figura 16.a.2).

Quando o segundo padrão  $\vec{b}_2$  (Figura 16.b.1) é apresentado, os neurônios talâmicos  $t_2$  e  $t_3$  vão disparar inicialmente sobre  $y_1$ , mediante as sinapses que foram reforçadas durante a apresentação de  $b_1$ . Em seguida a inibição de volta, também reforçada anteriormente conduzirá  $t_2$  e  $t_3$  ao estado de repouso (Figura 16. b.2).

Depois disto, e ainda durante a apresentação do segundo padrão  $\vec{b}_2$ , outro neurônio reticular  $y_2$  ganha a competição dentro da camada reticular. Assim (durante a apresentação do segundo padrão  $\vec{b}_2$ ,  $y_1$  diminui sua frequência de disparo porque  $t_2$  e  $t_3$  voltam ao repouso rapidamente. Como supomos que, neste instante, a primeira derivada da ativação total em  $y_2$  é maior que em  $y_1$ ,  $y_2$  ganha a competição na camada reticular. A probabilidade do neurônio reticular  $y_2$  ser ativado, como consequência da ativação dos inputs  $b_2^4$  e  $b_2^5$ , é calculada como a probabilidade do padrão  $\vec{b}_2$  após a inibição dos inputs  $b_2^2$  e  $b_2^3$  desde  $y_1$ :

$$P(\vec{y}_2) = P(\vec{b}_2) - P(\vec{b}_1 / \vec{y}_1)P(\vec{y}_1) \quad (89)$$

que é a segunda iteração do algoritmo de Gram-Schmidt. As conexões inibitórias  $P(\vec{b}_2^2 / \vec{y}_2)$  e  $P(\vec{b}_2^3 / \vec{y}_2)$  que partem de  $y_2$  para  $t_4$  e  $t_5$  começam a reforçar-se neste momento. Estas projeções inibitórias, que vão incrementando seus pesos, fazem que, finalmente,  $t_4$  e  $t_5$  voltem ao estado de repouso, da mesma maneira como aconteceu previamente com  $t_2$  e  $t_3$ . Em

conseqüência, pelo fato de  $y_2$  ter as entradas cada vez menos ativas,  $y_2$  diminui a sua ativação, da mesma forma como aconteceu antes com  $y_1$  (veja a parte superior da Figura 16.b.3). Em resumo, a queda rápida do valor das entradas de  $y_1$  e, posteriormente das entradas de  $y_2$ , inativa primeiro  $y_1$  e, depois,  $y_2$ . Se, numa etapa posterior, um outro padrão  $b_3$  fosse fornecido, por causa do mesmo processo, ativariam-se primeiro  $y_1$  e  $y_2$ , no caso de  $b_3$  ter componentes sobre estes eixos (ou neurônios reticulares) e, posteriormente, após ficar inativos, um terceiro neurônio  $y_3$  encarregaria-se da parte do padrão não representada nos eixos  $y_1$  e  $y_2$ .

***Introdução: A rede talâmica: uma rede neural de extração de componentes principais***

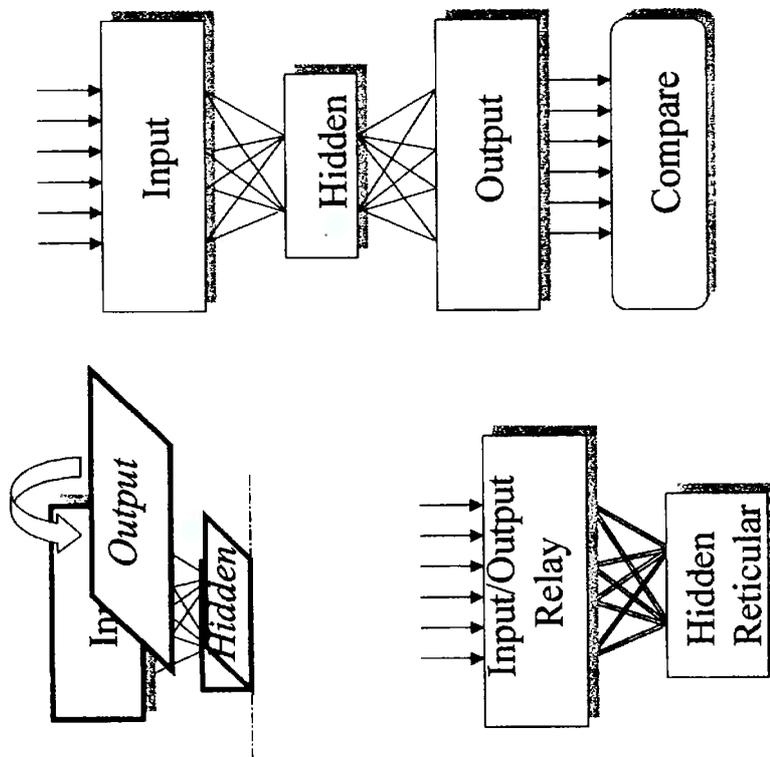
Nas seguintes seções será mostrado como a rede apresentada nas seções anteriores não somente gera um conjunto de eixos perpendiculares sobre os quais podem-se representar os padrões de entrada a ela, como também gera um conjunto de eixos que são os componentes principais do conjunto de padrões. Ou seja cada disparo de cada neurônio da camada reticular é um evento-vetor alinhado na direção de cada um dos componentes principais que são as direções de maior variabilidade dos padrões de entrada.

***Redes auto-associativas, uma primeira aproximação para entender a extração de componentes principais na rede talâmica.***

Para obter uma melhor compreensão da computação no tálamo, estudaremos a rede de tipo auto-associativa [3] para redução de dimensão mediante extração de componentes principais, PCA.

As redes lineares auto-associativas com poucos neurônios na camada escondida em comparação com as outras camadas realizam uma extração de componentes principais. A pequena camada escondida comprime os dados de entrada num espaço de dimensão menor. Cada neurônio da camada escondida representa um eixo deste espaço de menor dimensão, enquanto que os pesos entre a camada de entrada e um determinado neurônio da camada escondida, representam os componentes deste neurônio-eixo em função da base de vetores da entrada, a qual tem maior dimensão. Um módulo de comparação computa a diferença entre a saída e o padrão desejado (também chamado de *target*), o qual é igual ao padrão de entrada, permitindo a modificação dos pesos. Se dobrarmos este sistema como na Figura 17.1 obteremos a arquitetura neural do tálamo [16, 19] na qual as camadas de entrada e saída estão juntas numa única camada chamada camada *relay* formada pelos neurônios tálamo-corticais. O módulo de comparação também está encravado nesta camada *relay*. A camada intermediária que atua como uma dobradiça quando a rede é dobrada, chama-se camada reticular [22]. Ainda continuam existindo todas as conexões da rede auto-associativa, a única diferença é que as conexões desde a camada reticular à camada *relay* são negativas. Este sinal facilita a comparação entre a entrada e a saída e

contribui à modificação dos pesos.



**Figura 17** A arquitetura do tálamo é a mesma que poderia ser obtida dobrando-se uma rede auto-associativa. Os componentes principais são obtidos na camada reticular do tálamo, como acontece na camada tipo “pescoço de garrafa” da rede auto associativa.

### *Componentes principais para otimizar a compressão de dados*

A técnica dos componentes principais permite a compressão de dados para, por exemplo, poder enviá-los através de um canal de comunicações segundo se mostra na Figura 18. O passo de  $\bar{x}$  a  $\bar{y}$  realiza-se a través de uma transformação linear representada pela matriz de compressão  $C$  de tal forma que  $\bar{y} = C\bar{x}$ . Utilizando-se esta transformação linear o padrão de entrada  $\bar{x}$ , na figura 18, passa de um espaço de dimensão 14 para estar representado num espaço de 4 dimensões. Nesta transformação linear cada uma das 4 filas de  $C$  representa as coordenadas dos quatro vetores da base do novo espaço em função dos 14 vetores da base do espaço original.



**Figura 18.** Compressão de dados para serem enviados através de um canal de comunicações. Espera-se que após a descompressão da informação enviada pelo canal de comunicações o padrão reconstruído  $X'$  seja o mais parecido possível ao padrão inicial  $X$ .

Após a compressão e descompressão não é possível recuperar perfeitamente o vetor  $\bar{x}$ . Ao invés disto obteremos um vetor  $\bar{x}$  um pouco alterado  $\bar{x}'$ . Demonstra-se que para conseguir o máximo parecido de  $\bar{x}'$  com  $\bar{x}$  [6 página 401], as filas da matriz  $C$  devem ser os auto-vetores  $\phi_i$  correspondentes aos 4 maiores auto-valores  $\lambda_i$  da matriz  $R$ , que é a matriz de correlação dos padrões

de entrada  $\vec{x}$ , de modo que, pela própria definição de auto-vetores:

$$R\Phi = \Phi\Lambda \text{ onde } \Phi = [\vec{\phi}_1, \vec{\phi}_2, \dots, \vec{\phi}_n] \text{ e } \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \dots \\ & & & \lambda_n \end{bmatrix} \quad (90)$$

expressão esta que pode ser escrita assim:

$$R = \Phi\Lambda\Phi^{-1} \quad (91)$$

ou, tendo em vista que o módulo de cada auto-vetor é igual a 1:  $\Phi^T\Phi = I$  e  $\Phi^T = \Phi^{-1}$ . Portanto a matriz de correlação pode, de igual modo, se escrever da seguinte forma:

$$R = \Phi\Lambda\Phi^T \quad (92)$$

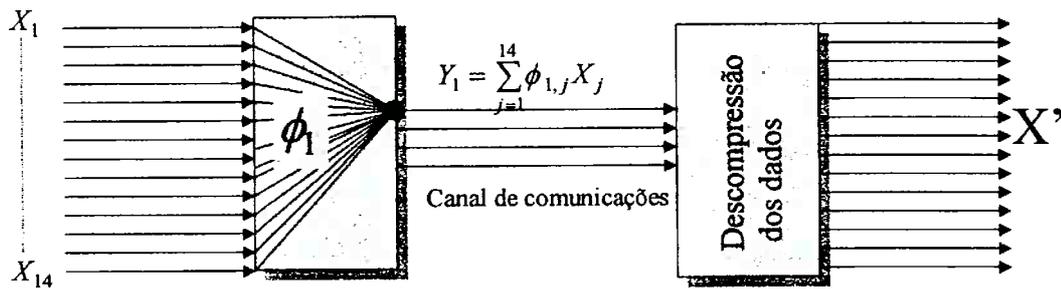
sendo a matriz dos auto-valores:

$$\Phi^T R \Phi = \Lambda \quad (93)$$

Deste modo a transformação linear  $\vec{Y} = C\vec{X}$  pode ser escrita também assim  $\vec{Y} = \Phi^T\vec{X}$ , de modo que, por exemplo, o sinal pelo primeiro canal de comunicação será :

$$Y_1 = \vec{\phi}_1^T \vec{X} \quad (94)$$

que corresponde ao produto escalar do vetor  $\vec{\phi}_1$  e o padrão  $\vec{x}$ :  $\phi_{1,1}X_1 + \phi_{1,2}X_2 + \dots + \phi_{1,14}X_{14}$ . Cada um dos fatores  $\phi_{1,j}$ , que são os componentes na base do espaço dos padrões de entrada do primeiro auto-vetor, podem também ser considerados como os pesos de um neurônio. A soma de produtos das entradas multiplicados pelos pesos, permite calcular o produto escalar anterior como pode ser observado na seguinte figura:



**Figura 19.** Os componentes do auto-vetor  $\vec{\phi}_1$  que multiplicado escalarmente pelo vetor  $X$  gera um dos componentes principais, poderiam ser consideradas como pesos de uma rede neural que relaciona uma entrada (neste caso de 14 componentes) com uma saída (neste caso de quatro componentes).

Portanto a matriz  $C$  pode ser implementada por uma rede neural onde os neurônios da segunda camada computam a soma das entradas ponderadas pelos pesos. Cada vetor formado pelos pesos de cada neurônio representam um auto-vetor.

### ***Condições para que os pesos evoluam na direção dos componentes principais***

Até agora temos visto que é possível dar a um conjunto de pesos de uma rede neural uns valores  $\phi_{i,j}$  de forma que a própria rede represente uma transformação linear de um espaço de dimensão elevada para um espaço de dimensão pequena, otimizando a recuperação do padrão de entrada na saída. Mas seria possível encontrar uma regra de atualização dos pesos de forma que, com computações restritas a cada neurônio, estes pesos possam evoluir em direção aos auto-vetores?

Isto é possível se:

- (a) *Os pesos evoluem em direção a valores estáveis.*

(b) Os disparos dos neurônios da camada de saída são produzidos em instantes distintos.

A afirmação anterior será demonstrada a seguir:

Chamaremos  $\bar{w}_i$  ao vetor de pesos em direção a um determinado neurônio após a convergência e estabilização dos pesos:

$$\Delta \bar{w}_i(n) \rightarrow 0 \text{ quando } n \rightarrow \infty .$$

Em conseqüência o valor limite para a saída dos neurônios (na segunda camada, ou no canal de comunicações) é:

$$\lim_{n \rightarrow \infty} y_i(n) = \bar{w}_i^T X(n) \quad (95)$$

Seja  $Y_i(n)$  uma variável aleatória cuja efetiva realização é dada pelo valor da saída  $y_i(n)$ . A correlação cruzada entre as variáveis aleatórias  $Y_i(n)$  e  $Y_k(n)$  em equilíbrio é:

$$\lim_{n \rightarrow \infty} E[Y_j(n)Y_k(n)] = E[\bar{w}_j^T X(n)X^T(n)\bar{w}_k] = \bar{w}_j^T R\bar{w}_k = \begin{cases} \lambda_i, & i = k \\ 0 & i \neq k \end{cases} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} = \Lambda \quad (96)$$

por causa dos disparos de cada neurônio acontecer em momentos distintos:

Onde R é a chamada matriz de auto-correlação. E para que possa acontecer  $\bar{w}_i^T R\bar{w}_j = \Lambda$  os  $\bar{w}_i$  devem ser iguais aos  $\bar{\phi}_i$ . Portanto, nessas condições, o valor do vetor de pesos de cada neurônio da camada de saída evolui até ficar igual a um dos auto-vetores.

Contudo ainda não foi explicada a regra exata de atualização dos pesos para conseguir que estes evoluam na forma anterior, cumprindo as condições a e b, de maneira que terminem ficando iguais aos auto-vetores. Esta explicação será fornecida nos seguintes parágrafos.

### ***Reforço hebbiano e componentes principais***

A regra de evolução dos pesos mais simples e a chamada regra de Hebb [8] que pode ser expressa matematicamente :

$$\Delta w_{ij} = \xi O_i I_j \quad (97)$$

onde  $I$  e  $O$  representam os disparos dos neurônios pré-sinápticos  $j$  e pós-sináptico  $i$  respectivamente, e  $\Delta w$  o incremento da eficiência sináptica, ou peso, entre os dois neurônios. Nos neurônios biológicos ou em aqueles de tipo probabilístico  $I$  e  $O$ , têm valor binário enquanto em outros tipos de neurônios artificiais  $I$  e  $O$  podem tomar outros valores numa gama contínua, por exemplo entre 0 e 1. Se tivermos uma rede de duas camadas com  $m$  neurônios na primeira camada e  $n$  neurônios na segunda, os pesos  $w_{i,j}$  nas sinapses que ligam um neurônio  $i$  da primeira camada a um neurônio  $j$  na segunda, poderiam ser expressos numa matriz de pesos  $W$ , de maneira que, os pesos de todas as sinapses sejam atualizados em cada iteração. Se identificarmos os disparos  $I_j$  dos neurônios da primeira camada com os valores dos componentes de um determinado padrão de entrada (por exemplo o padrão de entrada poderia ser uma imagem e os  $I_j$  os valores de uma escala de cinza no 'pixel'  $j$  desta imagem) fazendo uma identificação análoga com os neurônios da segunda camada, então os pesos entre neurônios poderiam medir as correlações entre os componentes do padrão de entrada e os componentes do padrão de saída (os pixels da imagem de entradas e os pixels da imagem na saída). Imagine-se que se deseja correlacionar vários pares de imagens e temos inicialmente todos os pesos

iguais a zero. Quando colocamos o primeiro par de padrões, um deles na entrada e o outro na saída, a seguinte matriz mostra os incrementos dos pesos em cada uma das sinapses.

$$W = \Delta W = \xi \begin{bmatrix} O_1 I_1 & O_1 I_2 \dots O_1 I_n \\ O_2 I_1 & O_2 I_2 \dots O_2 I_n \\ \vdots & \ddots \quad \vdots \\ O_n I_1 & O_n I_2 \dots O_n I_n \end{bmatrix} = \begin{bmatrix} O_1 \\ O_2 \\ \vdots \\ O_n \end{bmatrix} [I_1, I_2, \dots, I_n] \quad (98)$$

Se colocarmos um segundo par de padrões um deles na entrada e outro na saída, deve-se somar à matriz de incrementos anterior uma segunda matriz de incrementos, da seguinte forma (os índices superiores indicam cada um dos vários padrões de entrada-saída):

$$W = \Delta W^1 + \Delta W^2 = \xi \begin{bmatrix} O_1^1 \\ O_2^1 \\ \vdots \\ O_n^1 \end{bmatrix} [I_1^1, I_2^1, \dots, I_n^1] + \xi \begin{bmatrix} O_1^2 \\ O_2^2 \\ \vdots \\ O_n^2 \end{bmatrix} [I_1^2, I_2^2, \dots, I_n^2] \quad (99)$$

Para  $n$  pares de padrões de entrada-saída a matriz de pesos é:

$$W = \xi \cdot \begin{bmatrix} O^1, O^2, \dots, O^n \\ I^{1T} \\ \vdots \\ I^{nT} \end{bmatrix} = \xi \cdot O I^T \quad (100)$$

onde os vetores  $O^1, O^2$  etc, representam, em colunas o primeiro, segundo, etc. padrões de saída e os  $I^{1T}, I^{2T}$  etc., representam em filas o primeiro, segundo, etc. padrões de entrada.

A matriz  $W = O I^T$  foi obtida para a situação chamada de “aprendizado supervisionado” no qual são fornecidos pares de padrões, um deles na entrada e outro deles na saída da rede. O mesmo tipo de rede serve para um outro tipo de aprendizado chamado não supervisionado no qual não se

fornecem os padrões de saída. Os padrões de saída são o próprio resultado da rede.

$$O = WI \quad (101)$$

Portanto os pesos poderiam ser obtidos substituindo-se O em  $w = OI^T$ , resultando:

$$W = \xi OI^T = \xi WI I^T \quad (102)$$

Para que este processo não supervisionado conduza à extração de componentes principais é preciso que na anterior equação, ao invés do produto  $I \cdot I^T$  tenhamos o valor esperado ou média deste produto  $E(I \cdot I^T)$  o qual é a matriz de correlação R. Este valor esperado pode ser obtido supondo-se que os padrões de entrada já estão normalizados, após dividir cada componente do padrão de entrada I pela cardinalidade,  $n(I)$ , de cada padrão. Deste modo a matriz de pesos será:

$$W = \xi W \cdot E(I I^T) = \xi WR \quad (103)$$

Transpondo-se a expressão anterior e tendo-se em vista que por R ser simétrica,  $R^T = R$  :

$$\begin{aligned} W^T &= \xi(WR)^T = \xi R^T W^T \\ W^T &= \xi R W^T \\ \frac{1}{\xi} W^T &= R W^T \end{aligned} \quad (104)$$

considerando a seguinte matriz diagonal:

$$\Lambda = \begin{bmatrix} \frac{1}{\xi} & & & \\ & \frac{1}{\xi} & & \\ & & \ddots & \\ & & & \frac{1}{\xi} \end{bmatrix} \quad (105)$$

obteremos a condição para conseguir os auto-vetores da matriz R:

$$\Lambda W^T = RW^T \quad \therefore \quad \Phi = W^T \quad (106)$$

Onde,  $\Phi$ , a matriz de auto-vetores é igual à transposta da matriz de pesos. Portanto cada fila da matriz de pesos, ou seja todos os pesos de um mesmo neurônio de saída, é um auto-vetor. Como os auto-vetores da matriz de auto-correlação fornecem os componentes principais, os pesos de cada neurônio da camada de saída são os próprios componentes principais.

Contudo, neste caso, todos os auto-valores são iguais o que significa que todos os neurônios da camada de saída estão obtendo o mesmo componente principal: o primeiro. O leitor pode se perguntar: Porque o primeiro componente e não outro? Se a nossa rede só tivesse um neurônio na saída, teria-se exatamente o mesmo resultado, o mesmo autovalor e o mesmo auto-vetor, igual ao auto-valor e auto-vetor múltiplo do caso geral. Neste caso, com um único neurônio, este auto-vetor é certamente o primeiro.

Em resumo, vimos que os pesos de um neurônio evoluem em direção ao primeiro componente principal se são treinados pela regra de Hebb.

Se, ao invés de utilizarmos esta regra de Hebb, utilizarmos a regra de atualização de pesos que foi proposta para a rede talâmica.

$$w_{ij} = P(O_i / I_j) \quad (107)$$

então os disparos binários dos neurônios na camada de saída podem-ser escritos da maneira seguinte em função dos disparos binários na camada de entrada :

$$O = WI$$

$$\begin{bmatrix} O_1 \\ O_2 \\ \vdots \\ O_n \end{bmatrix} = \begin{bmatrix} P(O_1/I_1) & P(O_1/I_2) \dots P(O_1/I_n) \\ P(O_2/I_1) & P(O_2/I_2) \dots P(O_2/I_n) \\ \vdots & \ddots \vdots \\ P(O_n/I_1) & P(O_n/I_2) \dots P(O_n/I_n) \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} n(O_1 \cap \frac{I_1}{n(I_1)}) & n(O_1 \cap \frac{I_2}{n(I_2)}) \dots n(O_1 \cap \frac{I_n}{n(I_n)}) \\ n(O_2 \cap \frac{I_1}{n(I_1)}) & n(O_2 \cap \frac{I_2}{n(I_2)}) \dots n(O_2 \cap \frac{I_n}{n(I_n)}) \\ \vdots & \ddots \vdots \\ n(O_n \cap \frac{I_1}{n(I_1)}) & n(O_n \cap \frac{I_2}{n(I_2)}) \dots n(O_n \cap \frac{I_n}{n(I_n)}) \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} =$$

$$= \begin{bmatrix} \sum_{k=1}^K O_1^k i_1^k & \sum_{k=1}^K O_1^k i_2^k & \dots & \sum_{k=1}^K O_1^k i_n^k \\ \sum_{k=1}^K O_2^k i_1^k & \sum_{k=1}^K O_2^k i_2^k & \dots & \sum_{k=1}^K O_2^k i_n^k \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{k=1}^K O_n^k i_1^k & \sum_{k=1}^K O_n^k i_2^k & \dots & \sum_{k=1}^K O_n^k i_n^k \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = O_i^T \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = O_i^T I = WI$$

(108)

onde o índice k que aparece na matriz grande permite somar os disparos binários simultâneos em cada uma das sinapses que unem as entradas com as saídas ao longo das apresentações dos K padrões.

É preciso recordar aqui, que esta rede talâmica, é uma rede de tipo probabilística. As entradas e as saídas são binárias. Embora estas entradas e saídas sejam binárias acontecem segundo uma determinada probabilidade. Por exemplo, se o padrão de entrada for uma imagem e o brilho de cada *pixel* estive-se representado numa escala de cinza indo desde 0 até 1, esta escala será codificada com um disparo do neurônio cuja probabilidade de disparo tem um range de zero até 1.

Os neurônios de saída disparam segundo a seguinte matriz de probabilidades:

$$\begin{bmatrix} P(O_1) \\ P(O_2) \\ \vdots \\ P(O_n) \end{bmatrix} = \begin{bmatrix} P(O_1/I_1) & P(O_1/I_2) \dots P(O_1/I_n) \\ P(O_2/I_1) & P(O_2/I_2) \dots P(O_2/I_n) \\ \vdots & \ddots \quad \vdots \\ P(O_n/I_1) & P(O_n/I_2) \dots P(O_n/I_n) \end{bmatrix} \begin{bmatrix} P(I_1) \\ P(I_2) \\ \vdots \\ P(I_n) \end{bmatrix} \quad (109)$$

Será demonstrado agora que, a anterior modelagem estatística dos pesos, permite, como acontecia com a regra clássica de Hebb, a obtenção dos componentes principais.

A matriz de pesos (equação 108) que era:

$$\begin{aligned} W &= Oi^T \\ W &= WI^T \\ W^T &= I^T iW^T = \begin{bmatrix} \gamma_{n(I_1)} & & & \\ & \gamma_{n(I_2)} & & \\ & & \ddots & \\ & & & \gamma_{n(I_n)} \end{bmatrix} I I^T W^T \\ & \begin{bmatrix} n(I_1) & & & \\ & n(I_2) & & \\ & & \ddots & \\ & & & n(I_n) \end{bmatrix} W^T = I I^T W^T \end{aligned} \quad (110)$$

Se cada um dos padrões  $I_k$  houver sido apresentado o mesmo número de vezes para a rede tendo um total de  $K$  padrões diferentes, obteríamos a seguinte matriz dividindo por este valor  $K$  ambos membros da equação anterior:

$$W = O_i^T$$

$$\begin{bmatrix} n^{(I_1)}/K \\ n^{(I_2)}/K \\ \dots \\ n^{(I_n)}/K \end{bmatrix} W^T = \frac{1}{K} I I^T W^T$$

ou seja:

$$\Lambda W^T = R W^T \quad (111)$$

E portanto, observando-se a última equação, pode-se comprovar que, com esta modelagem dos pesos sinápticos, cada fila da matriz  $W$  representa os componentes principais na base dos padrões de entrada.

$$\Phi = W^T \quad (112)$$

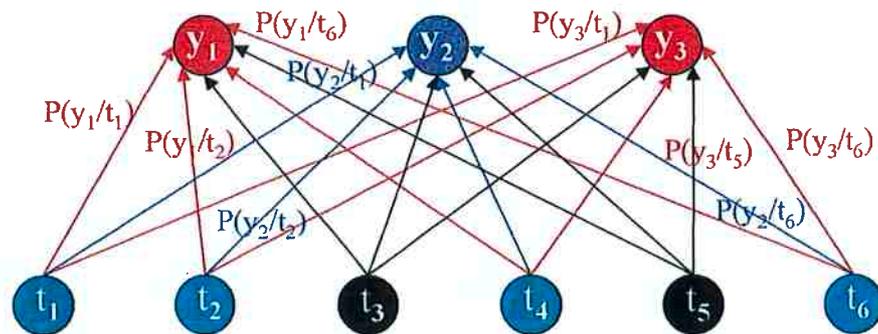
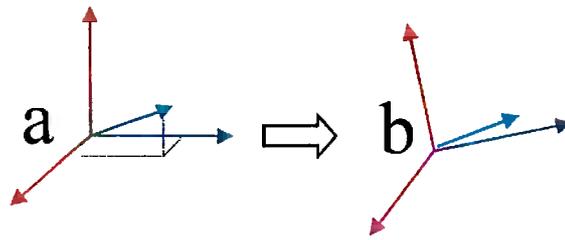
Peguemos a primeira fila  $[P(O_1/I_1) \ P(O_1/I_2) \dots P(O_1/I_n)]$ . Sabemos ( ver página 14) que cada uma dos componentes deste vetor representa uma projeção, por exemplo:

$$P(O_i / I_i) = n(\bar{O}_i \cap \bar{i}_i) \quad (113)$$

Contudo, neste caso de atualização estatística dos pesos, acontece o mesmo que acontecia no caso anterior de atualização *Hebbiana*: Só é possível obter-se um único componente principal, o primeiro componente, em todos os neurônios da camada de saída. Para obter-nos mais componentes principais, em primeiro lugar, deve-se garantir que as outras possíveis direções sejam ao menos ortogonais entre elas. Foi demonstrado anteriormente, que convergindo-se os pesos para valores estáveis, e sendo a segunda camada, uma camada competitiva, os pesos que chegam a cada neurônio vão convergindo em direção aos auto-vetores. Contudo, é possível uma convergência mais rápida se cada vez que se alterem os pesos de um

neurônio, chamemo-lo neurônio A, em direção a um determinado componente principal, o resto dos pesos dos outros neurônios sejam modificados para que, ao menos, representem vetores de um sub-espaço perpendicular ao vetor de pesos do neurônio A. Se, por exemplo, os pesos de cada neurônio tivessem sido inicializados, de forma que representem vetores mutuamente ortogonais mas sem serem ainda os componentes principais, esta modificação dos pesos permitirá girar em cada momento todo o conjunto de eixos de forma que em cada momento seja mantida a ortogonalidade enquanto alguns dos eixos se aproximam aos componentes principais.

Na Figura 20 mostra-se a estrutura de dupla camada da rede talâmica. Lembre-se que a segunda camada é uma camada competitiva. Só um neurônio, o de maior incremento de ativação, dispara por vez. Suponhamos portanto que o neurônio  $y_2$  da segunda camada, dispara quando é apresentado o padrão de entrada constituído pelos disparos simultâneos dos neurônios  $t_1$ ,  $t_2$ ,  $t_4$  e  $t_6$ , representados em azul. Os pesos das sinapses de neurônio  $y_2$ , procedentes de  $t_1$ ,  $t_2$ ,  $t_4$  e  $t_6$  são incrementados enquanto os pesos de  $y_2$  procedentes dos neurônios inativos  $t_3$  e  $t_5$  ficam iguais. Na parte superior da figura mostra-se um conjunto de vetores, o vetor azul representando o padrão de entrada. Mostram-se também três eixos ortogonais representando os pesos dos neurônios  $y_1$ ,  $y_2$  e  $y_3$ . O vetor de pesos do neurônio ganhador  $y_2$  está em violeta e recebe a maior projeção desde o padrão de entrada.



**Figura 20** Esquema de conexões entre os neurônios tálamo-corticais e os neurônios reticulares. O padrão de entrada está representado pelos neurônios tálamo-corticais ativos (em azul). Os neurônios tálamo-corticais em preto estão inativos. Aparecem nas conexões os valores dos pesos sinápticos. O neurônio reticular violeta está disparando enquanto os neurônios em laranja estão em silêncio. Por causa dos pesos serem probabilidades condicionais os pesos das conexões em violeta aumentam, os das conexões em laranja diminuem e os pesos das conexões em preto não se alteram.

O aumento dos pesos desde  $t_1$ ,  $t_2$ ,  $t_4$  e  $t_6$  significa que a projeção geométrica desde o vetor  $y_2$ , o violeta, ao padrão de entrada, o azul vão aumentar, ou em outras palavras, o vetor violeta vai-se aproximar do vetor azul. Por causa deste giro do vetor violeta, os outros eixos, os laranjas, deixariam de ser ortogonais ao vetor violeta. Contudo, isto não acontece porque as sinapses, as conexões laranjas, entre os neurônios azuis representando o padrão de entrada e os neurônios laranjas, inativos, vão diminuir seus pesos. Isto acontece porque o denominador das probabilidades condicionais aumentam

enquanto o numerador fica igual. A diminuição dos pesos é equivalente a uma diminuição das projeções do vetor azul, o padrão de entrada, sobre os eixos laranjas. Como o vetor azul não se move, a única forma de diminuir as projeções ditas é movimentando-se os eixos laranjas para que fiquem ortogonais ao vetor azul. Pode-se observar portanto, que cada padrão de entrada leva o eixo de maior projeção na sua própria direção, enquanto os outros eixos são afastados para que fiquem ortogonais a ele. Este resultado acontece por causa da competitividade entre os neurônios da segunda camada (camada reticular), e por causa do tipo de reforço entre neurônios baseado em probabilidades condicionais.

No gráfico b da Figura 20 mostra-se que o vetor violeta se aproxima do vetor azul mas não chega a coincidir com ele. Se o vetor azul fosse o único padrão de entrada, e fosse apresentado durante bastante tempo para a rede, o eixo violeta se aproximaria cada vez mais ao vetor azul enquanto os eixos laranjas seriam cada vez mais ortogonais a este vetor azul. Isto não acontece quando são apresentados mais padrões. Com mais padrões, cada um deles consegue disparar o neurônio com pesos mais próximos a eles, e em consequência, os componentes principais vão sendo orientados pelos padrões de entrada. Finalmente chega-se a um equilíbrio onde os eixos estão praticamente estáveis. Uma das razões deste equilíbrio é que os 'pulos' que vai dando todo o conjunto de eixos, uns eixos a favor e outros em contra dos padrões que vão-se apresentando, são cada vez mais pequenos. Isto é devido os denominadores das probabilidades condicionais serem cada vez maiores.

Uma vez treinado o sistema anterior, e já durante a etapa de teste, obteríamos só o primeiro componente principal para cada padrão de teste quando o

neurônio da camada reticular, com pesos mais próximos ao padrão de teste dispara. Obviamente isto é uma grande limitação pois deseja-se ter, mediante a rede, uma descrição completa de cada padrão em função dos componentes principais.

### ***Competitividade e recorrência para obter componentes principais distintos.***

Para conseguir obter mais de um componente principal para cada padrão é preciso colocarmos as conexões inibitórias recorrentes que, para maior clareza da explicação, foram omitidas anteriormente.

Foi mostrado na página 96 que estas conexões recorrentes desempenhavam um papel muito importante na computação de Gram-Schmidt na rede talâmica. Nesse caso as conexões inibitórias recorrentes subtraíam do padrão de entrada, o componente deste padrão de entrada sob cada um dos eixos considerados em instantes anteriores. Nesse caso acontecia uma simples extração de eixos ortogonais. Cada padrão apresentado para a rede permanecia na entrada da rede até que os pesos se estabilizassem, e as inibições da segunda camada sobre a primeira anulassem completamente as entradas. Por causa disso cada padrão de entrada à rede, se tinha algum elemento diferencial em relação aos padrões de entrada anteriores, era responsável pela geração de um novo eixo. Contudo estes eixos não eram os componentes principais. Para obter-se os componentes principais, cada padrão é apresentado durante menos tempo à rede, e todos os padrões são apresentados várias vezes à rede antes dos pesos começarem a se estabilizar.

Neste caso, quando um padrão é apresentado, acontece uma competição entre os neurônios da camada reticular. O neurônio ganhador é aquele cujos pesos se parecem mais ao padrão de entrada. Além disto os pesos deste neurônio vão representar algum dos futuros componentes principais, uma vez todo o universo dos padrões de entrada seja apresentado para a rede. Quando

o neurônio ganhador  $y_i$  dispara, ativam-se as projeções inibitórias recorrentes que mediante os pesos  $P(\bar{B}/\bar{y}_i)$  subtraem ao padrão de entrada, chamemo-lo B, a sua própria projeção  $n(B \cap \bar{y}_i)$  sobre o componente principal representada pelo disparo do neurônio ganhador  $y_i$ . Deste modo o resto do padrão B fica num sub-espaço ortogonal ao eixo representado pelo neurônio ganhador  $y_i$ . Já foi visto que os neurônios  $y_i$  da segunda camada, a camada reticular, disparam exclusivamente enquanto sua ativação (soma do produto das entradas pelos pesos) é crescente. Contudo neste momento a ativação do neurônio  $y_i$  começa a decrescer por causa da inibição recorrente do próprio neurônio  $y_i$  sobre as suas próprias entradas. Por isto o neurônio  $y_i$  não vai ser ativado pela fração de B que resta após a subtração da projeção de B sobre  $y_i$ .

O processo de extração de eixos começa de novo com esta fração restante do padrão B depois da inibição recorrente. O novo neurônio ganhador será o neurônio que representa a segunda maior projeção ou, em outras palavras, será o seguinte componente principal do padrão. Este neurônio também vai disparar subtraindo do padrão de entrada este componente. Portanto o padrão de entrada fica sem os dois maiores componentes e todo o processo repete-se de novo com o resto do vetor B.

O método anterior de subtração iterativa dos componentes principais de cada padrão é explicado por uma característica dos neurônios, já demonstrada anteriormente, que consiste em extrair o primeiro componente principal de um conjunto de padrões. Em todo momento, cada neurônio extrai o primeiro componente principal, só que o segundo componente, por exemplo, seria equivalente ao primeiro componente num sub-espaço onde o primeiro componente foi subtraído.

### ***Demonstração do PCA no tálamo usando-se a estatística Euclidiana.***

Na pag. 36 e seguintes vimos uma demonstração bastante simples de um algoritmo para o cálculo dos componentes principais. Sugerimos ler de novo, com cuidado, aquelas páginas. Copiemos aqui de novo o algoritmo da pag. 38.

$$\vec{Y}_1 = \vec{P}$$
$$\vec{Y}_{r+1} = \Psi \left( \vec{P} - \sum_{i=1}^r \vec{P}_{\vec{Y}_i} \right)$$

Basicamente estas duas equações materializam-se na primeira das duas camadas do tálamo, a chamada camada de neurônios tálamo-corticais. O termo subtrativo da segunda equação é uma reconstrução do padrão  $\vec{P}_i$  apresentado na entrada. A apresentação de cada um dos padrões vai ativar seqüencialmente um grupo dos eixos  $\vec{Y}_i$ . O eixo  $\vec{Y}_i$  ativado primeiro é aquele cuja ativação provinda das entradas é a maior. Portanto, para a seleção de cada  $\vec{Y}_i$  em cada momento é utilizado um processo competitivo segundo o teorema de Bayes que já foi apresentado. (Lembre-se que a biologia consegue calcular o máximo mediante uma computação em paralelo sem precisar do típico cálculo em série que seria mais demorado (Ver pag.88). Observe-se que os pesos que chegam aos neurônios reticulares associam os desvios em relação à média (estando estes desvios apresentados no espaço dos neurônios relê) com a atividade dos próprios neurônios reticulares.

Enquanto o neurônio reticular de sub-índice 1, o  $\vec{Y}_1$ , é ativado inicialmente

pela grande maioria dos padrões  $\vec{P}_i$ , um neurônio reticular  $\vec{Y}_i$  com sub-índice alto vai ser ativado quando desvios (da média) muito específicos são encontrados. Estes desvios possivelmente apareçam em muito poucos padrões.

Por outra parte o feed-back inibitório desde a primeira componente,  $\vec{Y}_1$ , produzirá uma inibição pouco específica. Por exemplo, se os padrões apresentados para a rede fossem diversos rostos, a inibição provinda de  $\vec{Y}_1$  forneceria um rosto médio pouco definido, porque as conexões de feed-back de  $\vec{Y}_1$  em direção à primeira camada  $P(\vec{b}_1 / \vec{Y}_1)$  são reforçadas por uma grande parte dos rostos. Contudo o disparo de um  $\vec{Y}_i$  de sub-índice elevado produzirá uma inibição mais específica, pois possivelmente existam poucos padrões que possuam o elemento diferencial ou distintivo associado a esse  $\vec{Y}_i$ . Voltando ao exemplo dos rostos, o bigode do artista plástico Salvador Dalí, que é um elemento diferencial importante num rosto específico, ativaria um  $\vec{Y}_i$  de sub-índice elevado. O feed-back inibitório desde este  $\vec{Y}_i$  reconstruiria rapidamente o rosto do pintor por ser o rosto de Dalí o único rosto associado a este  $\vec{Y}_i$  específico.

Demonstrar-se-á em seguida que o algoritmo apresentado para o cálculo dos componentes principais é muito parecido ao “Generalized Hebbian Algorithm” (GHA) descrito por Haykin<sup>4</sup> [36, eq. 8.133 e 8.134] no qual o vetor incremento de peso (em negrito) de ida a um determinado neurônio reticular  $y_k$  é proporcional a sua ativação  $y_k$  e ao erro entre o padrão de entrada  $X$  e o padrão reconstruído  $X^*$ <sup>5</sup>:

<sup>4</sup> A notação das equações do livro de Haykin foram adaptadas à notação da tese.

<sup>5</sup> Este erro é calculado na primeira camada da rede talâmica pelos neurônios tálamo-corticais.

$$W_k(t+1) = W_k(t) + \eta \cdot y_k(t)[X(t) - X_k^*(t)] \quad (114)$$

onde o padrão reconstruído  $x^*$  (chamado “reestimator” ou re-estimador) é calculado em cada momento da seguinte maneira:

$$x_k^*(t) = \sum_{i=1}^k W_i(t)y_i(t) \quad (115)$$

se o re-estimador usa todos (os n) componentes principais ao invés das “j” componentes, esta última equação pode ser definida como:

$$x_k^*(t) = \sum_{i=1}^n W_i(t)y_i(t) \quad (116)$$

Tendo em vista que no Generalized Hebbian Algorithm não existem conexões recorrentes (ou de volta) o cálculo anterior independe da arquitetura neuronal do GHA. Se o GHA tivesse conexões de feed-back, os pesos de volta de um neurônio  $y_i$  em direção a os neurônios  $x_j$  seriam iguais aos pesos desde os  $x_j$  em direção aos  $y_i$ . No caso da rede talâmica estes dois tipos de pesos são diferentes.

Lembre-se que na rede talâmica, as inibições recorrentes criam uma imagem de espelho reconstruindo ou re-estimando o padrão de entrada segundo o teorema da probabilidade total (equação 7). Deste modo o padrão  $X(t)$ , após ter sido totalmente reconstruído pela rede talâmica na iteração  $t \rightarrow \alpha$  é:

$$X(t) = \sum_{i=1}^n P(X(t) / y_i) \cdot y_i \quad (117)$$

No caso do GHA, uma vez terminado o treinamento do GHA, o incremento de peso tende à zero, portanto:  $W_j(t+1) \approx W_j(t)$  ficando a equação 14 da

seguinte forma:

$$\begin{aligned} w_j(t+1) - w_j(t) &= \eta \cdot y_j(t) [x(t) - x_j^*(t)] \\ 0 &= \eta \cdot y_j(t) [x(t) - x_j^*(t)] \\ 0 &= x(t) - x_j^*(t) \\ x(t) &= x_j^*(t) \end{aligned}$$

Em consequência:

$$x_k(t) = \sum_{i=1}^k w_i(t) y_i(t) \quad (118)$$

onde, como foi visto na eq. 117, cada um dos  $w_j$  podem ser calculados segundo uma probabilidade condicionada. Portanto o cálculo dos pesos de “feed-back” ou retorno da rede talâmica corresponde ao cálculo dos pesos no GHA.

Contudo os pesos para frente ou “feed-forward” da rede talâmica são calculados como  $w_{ij} = P(y_i / x_j)$  ao invés de  $P(x_j / y_i)$  que na verdade correspondem aos  $w_{ji}$ . O cálculo dos  $w_{ij}$  é feito, em cada momento, segundo o teorema de Bayes onde, na realidade, os  $x_j$  devem ser substituídos pelos  $x_j^*$ ; calculados mediante o teorema da probabilidade total. Assim:

$$P(y_i / x(t+1)_j^*)_{t+1} = \frac{P(y_i \cap x(t)_j^*)}{P(x(t+1)_j^*)} = \frac{P(y_i \cap x(t)_j^*)}{\sum_{j=1}^k P(x(t+1)_j^* / y_i) P(y_i)} = \frac{P(y_i / x(t)_j^*) \cdot P(x(t)_j^*)}{\sum_{j=1}^k P(x(t+1)_j^* / y_i) P(y_i)} \quad (119)$$

Desta forma a rede talâmica é, na realidade, um algoritmo de re-estimação iterativa das probabilidades dos “antecedentes o causas”, os  $y_i$ , uma vez conhecida a estimação dos  $x_j$ .

É fácil relacionar a anterior equação com a arquitetura talâmica colocando os

$w_{ij}$  e os  $w_{ji}$  ao invés dos  $P(y_i / x_j)$  e os  $P(x_j / y_i)$ .

$$w_{ij}(t+1) = \frac{w_{ij}(t) \cdot P(x(t)_j^*)}{\sum_{j=1}^k w_{ji}(t+1) \cdot P(y_i)} \quad (120)$$
$$\therefore w_{ij}(t+1) = \frac{P(x(t)_j^*)}{P(x(t+1)_j^*)} \cdot w_{ij}(t)$$

Que significa que os pesos terminarão se estabilizando  $w_{ij}(t+1) \approx w_{ij}(t)$  quando o numerador e o denominador sejam iguais ou seja quando a probabilidade estimada de  $X_j$  no instante  $t$  y  $t+1$  sejam iguais.

### ***Conclusões da Parte III***

Nesta terceira parte foram utilizadas as equações da parte primeira para desenvolver um novo modelo de rede neural para extração de componentes principais. Esta rede neural está baseada na arquitetura e propriedades de uma estrutura existente no cérebro, o tálamo. Nesta tese não se pretende estudar como as propriedades biológicas do tálamo concordam com o modelo matemático. Mas bem pretendemos nos inspirar nesta estrutura cerebral para realizar um novo modelo de rede que, como foi mostrado, tem interessantes propriedades, como por exemplo a capacidade de extrair os componentes principais e a propriedade de ter as fases de treinamento e de teste fundidas numa fase só, o que não acontece com as outras redes.

## **PARTE IV. METODOLOGIA E IMPLEMENTAÇÃO**

### ***DESCRIÇÃO GERAL DO FUNCIONAMENTO DO PROGRAMA***

O programa da rede talâmica realizado em Matlab modela o funcionamento de duas camadas de 81 neurônios sendo uma delas a camada de entrada e a outra a camada de processamento que, no tálamo, se chama camada reticular. Nas figuras que seguem o potencial base de membrana de cada um dos neurônios da camada chamada reticular se representa como a altura de cada um dos nós da rede desenhada sobre a superfície do gráfico central. Pode ser observado que o potencial de membrana base de todos estes neurônios sobe e desce ao mesmo tempo originando oscilações, que segundo os neurocientistas são a base dos ritmos do eletroencefalograma. O sincronismo entre os neurônios é possivelmente conseguido por causa das conexões dendrodendríticas do tipo gap-junction que são conexões citoplasmáticas diretas sem a necessidade de neuro-transmissor. Isto faz que o potencial de uns neurônios se comunique para os outros.

As duas camadas de neurônios estão conectadas mediante 81x81 pesos sinápticos em direção a frente e outros 81x81 pesos sinápticos de volta. Cada um destes pesos é atualizado em cada iteração.

Com estas duas camadas propõe-se que o tálamo faça uma tarefa de obtenção dos chamados componentes principais dos padrões que são apresentados para a rede. Tendo estes componentes principais, a rede talâmica pode realizar a tarefa de reconstrução de padrões que se mostrará num exemplo posterior. Cada um dos componentes principais não são outra coisa além do

vetor de pesos que chegam a cada um dos neurônios da segunda camada ou camada reticular.

Inicialmente a rede foi treinada com várias letras e números: H, B, b, 4 e 3. Após 4 apresentações deste conjunto de caracteres alfanuméricos, a rede convergiu para cada um dos caracteres fornecidos. Embora, por motivos de modelagem biológica 81 neurônios foram colocados na segunda camada, só 6 deles atuaram realmente. O fato de colocar 81 neurônio foi para demonstrar o tipo de comportamento global dos neurônios do tálamo.

Nas figuras que seguem têm seis gráficos em volta do gráfico central. No lado esquerdo da figura o primeiro gráfico é o padrão que é fornecido em cada momento para a rede. O segundo gráfico é o padrão reconstruído pela rede mediante as conexões inibitórias de feed-back à primeira camada. O terceiro gráfico é a atividade dos neurônios da primeira camada, que na verdade é a subtração dos dois gráficos anteriores. O quarto gráfico (acima a direita) é o neurônio ganhador da camada reticular em cada momento (lembre-se que a camada reticular é uma camada de tipo competitivo). O quinto gráfico são os neurônios da camada reticular que, tendo sido ganhadores anteriormente, agora estão em período refratário. Finalmente o gráfico da direita abaixo avalia a importância de cada um dos neurônios que foram ativados na hora de inibir a primeira camada (lembremos que, embora o neurônio reticular tenha parado de disparar sua inibição fica durante um tempo posterior agindo sobre a primeira camada).

Para maior facilidade a explicação do funcionamento da rede será realizada com o exemplo seguinte.

### EXEMPLO DE FUNCIONAMENTO

Neste exemplo, fornecemos pela primeira vez o padrão “B” para que a rede, que já tinha sido treinada da mesma forma com os padrões “H” e “4”. Na Figura 21 o potencial da maior parte dos neurônios está num nível ainda baixo (Na verdade nos estamos fornecendo cada novo padrão sempre que o potencial dos neurônios está num nível relativamente baixo) . Ainda nenhum neurônio da segunda camada disparou e portanto nenhuma inibição de volta sobre a primeira camada teve tempo de acontecer.

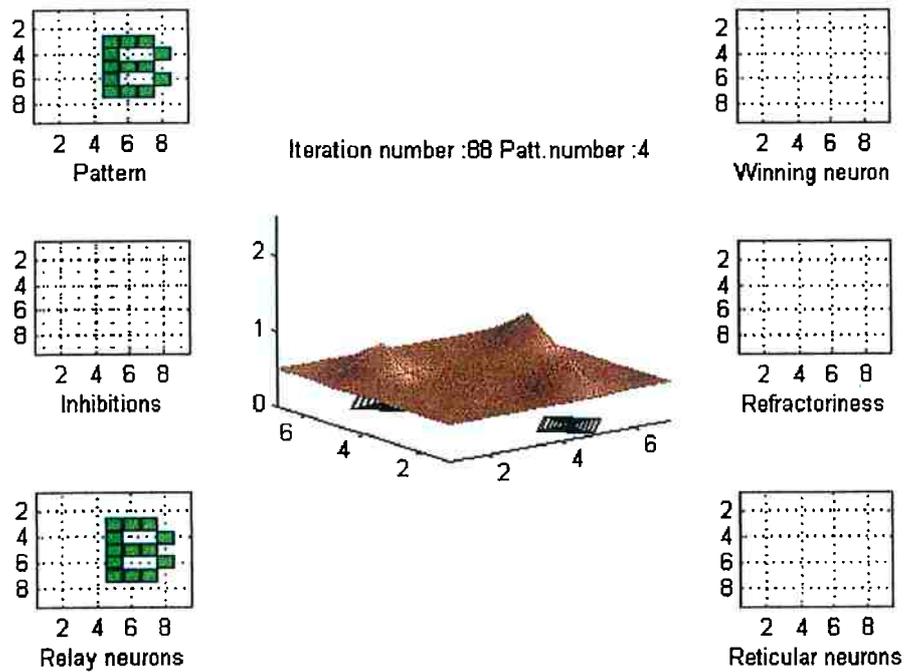


Figura 21

Na Figura 22 o neurônio (6,3) da segunda camada disparou e o potencial de base dos neurônios que aparece na figura central subiu levemente. A inibição procedente deste neurônio (6,3) não foi ainda o suficientemente forte para poder aparecer representada na figura.

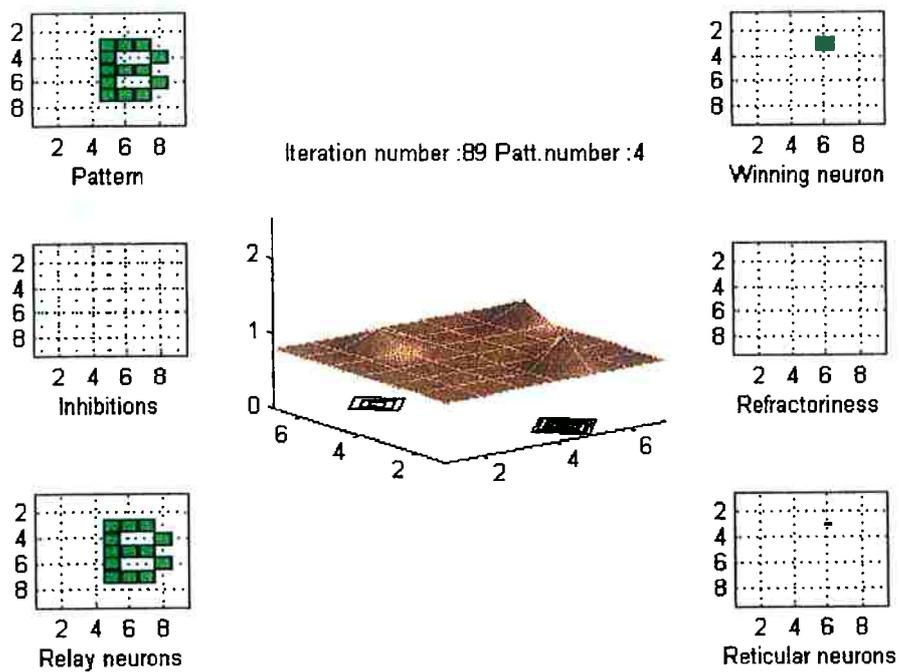


Figura 22

Na figura seguinte, Figura 23, o potencial inibitório desde o neurônio (6,3) aumentou e gerou a inibição com uma forma que é uma mistura dos dois padrões apresentados anteriormente: uma “H” e um “4”. Este primeiro neurônio terminará realizando uma inibição que será uma média de todos os padrões apresentados. O terceiro gráfico da esquerda que é a ativação dos neurônios “relay” da primeira camada, começa a ficar um pouco mais fraco após a inibição pelo neurônio (6,3).

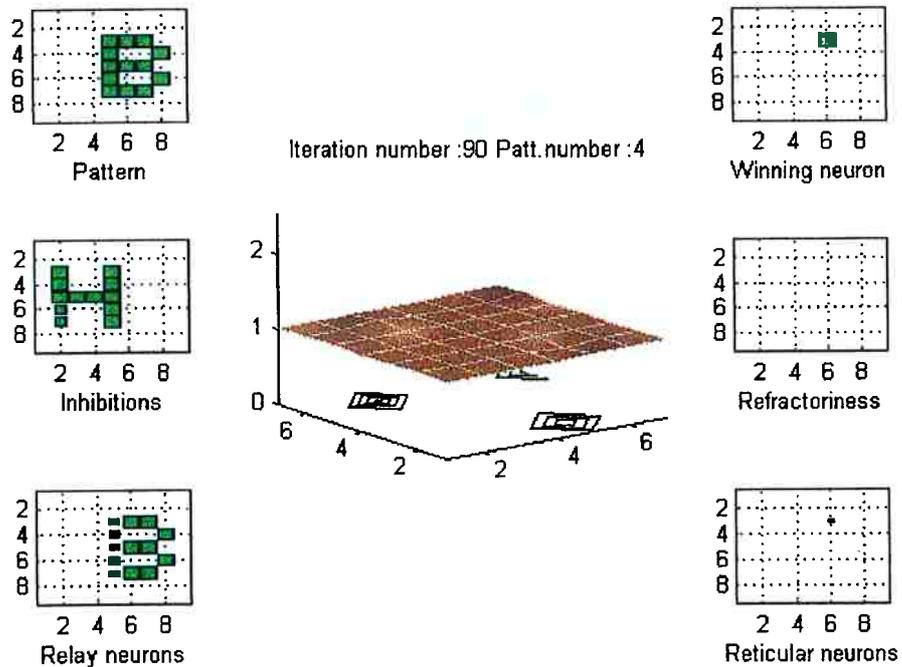


Figura 23

Na Figura 24 pode-se observar que o neurônio (6,3) entrou no estado refratário e um outro neurônio reticular disparou, o neurônio (4,7). Observe-se que na gráfica central o neurônio (6,3) diminuiu o seu potencial de base em relação aos outros neurônios enquanto que o (4,7) aumentou seu potencial devido ao reforço sináptico que está acontecendo com este último. Ainda a inibição tem a forma errada, bem diferente do objetivo que é da forma da letra “B”.

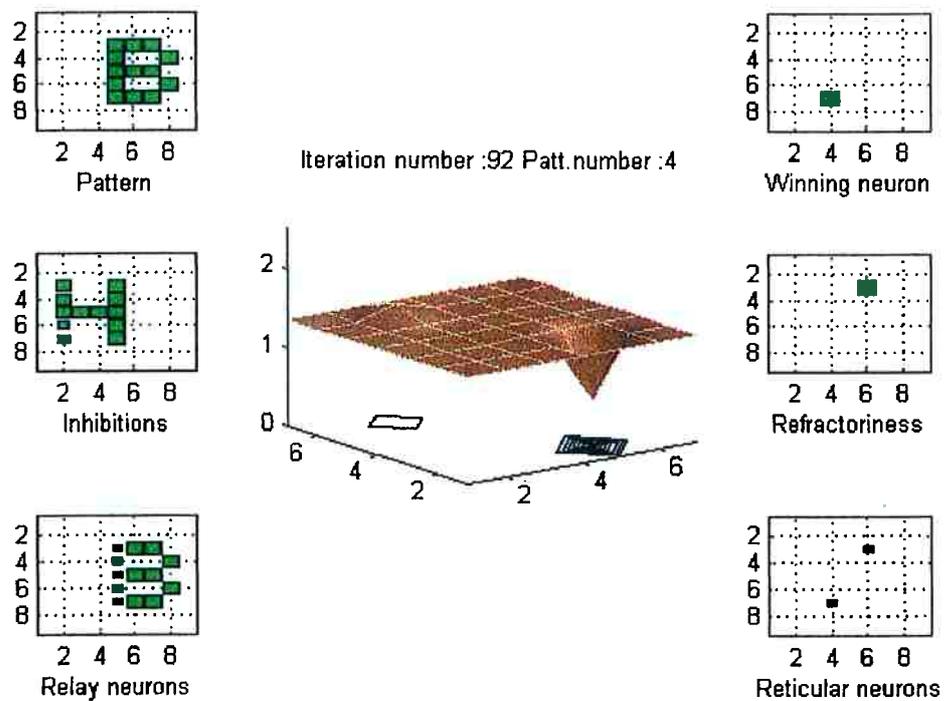


Figura 24

Na Figura 25 os dois neurônios ganhadores reduzem a sua atividade. Isto acontece porque a inibição de volta vai ficando maior em cada momento. A atividade dos neurônios “relay” ficou em forma de “3” devido a inibição sobre o padrão “B” na primeira camada. Observa-se que a inibição em forma de “H” é ainda bem diferente da que deveria ser (em forma de “B”).

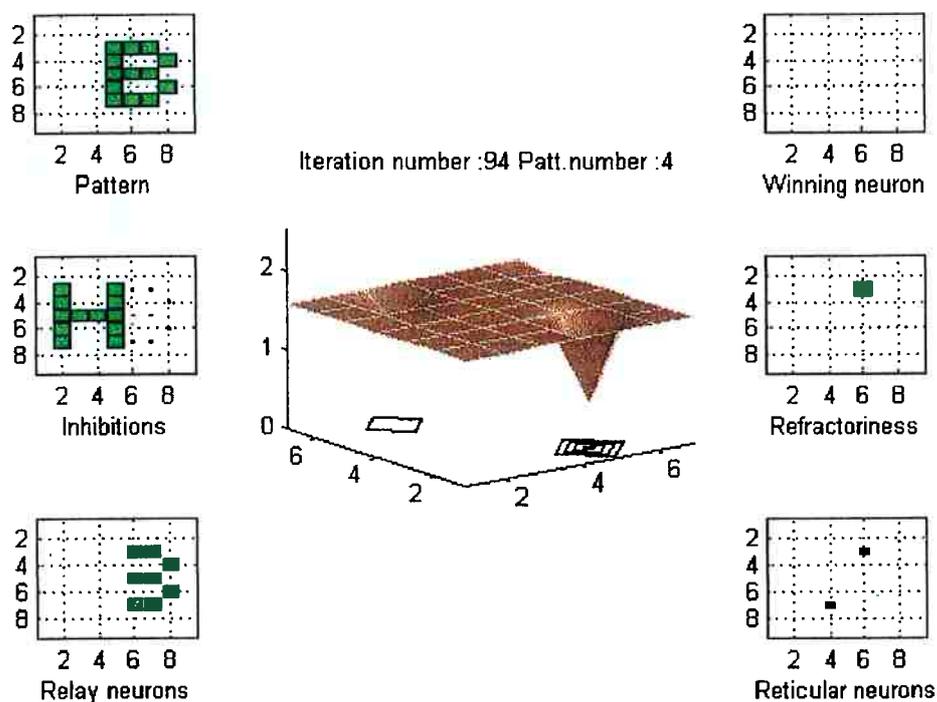


Figura 25

Na Figura 26 a inibição restante dos dois neurônios que já entraram em período refratário é fraca (em forma de uma “H”) e também é fraco o peso de algumas das conexões de “feed-back”. O terceiro neurônio ganhador (1,3) começa a “processar” a parte da “B” que o diferencia dos padrões anteriores (“4” e “H”) e que tem forma de “3”. Com as conexões de “feed-back” se reforçando desde o (1,3) já começa a aparecer a inibição objetivo em forma de “B”.

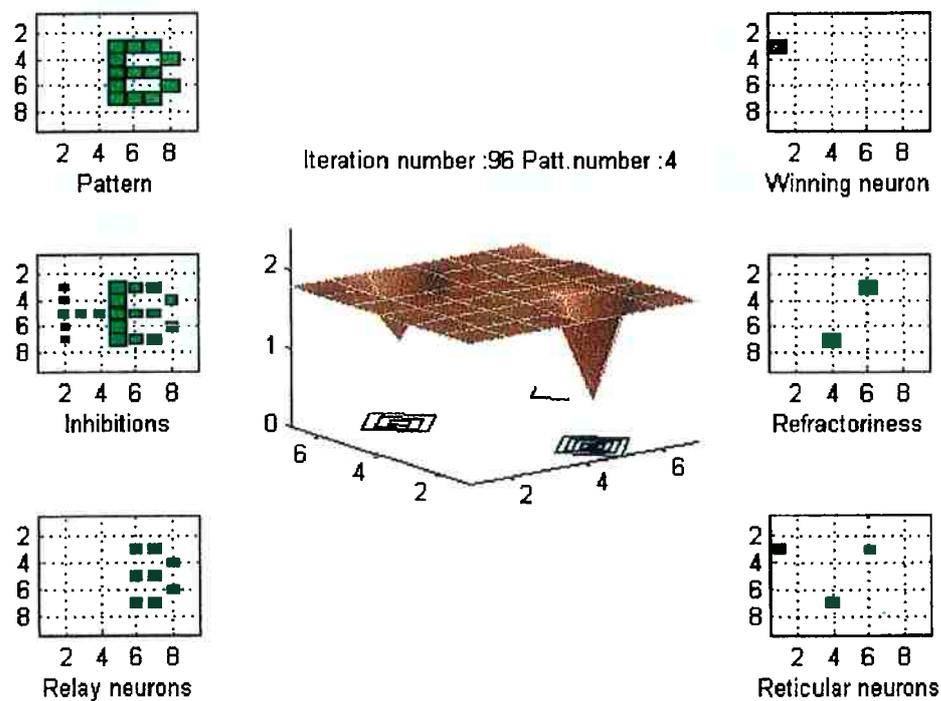


Figura 26

Na Figura 27 as conexões de volta desde o neurônio reticular (1,3) parecem ter-se reforçado o suficiente para junto com as inibições restante formar uma letra “B” na segunda figura da esquerda. Embora o objetivo de reproduzir a letra “B” com as inibições foi realizado, serão necessarias mais repetições da

letra “B “ para que a rede aprenda esta letra e seja capaz de completá-la se vier fragmentada ou deteriorada. Após esta inibição pode observar-se que a ativação dos neurônios “relay” da primeira camada é muito mais fraca.

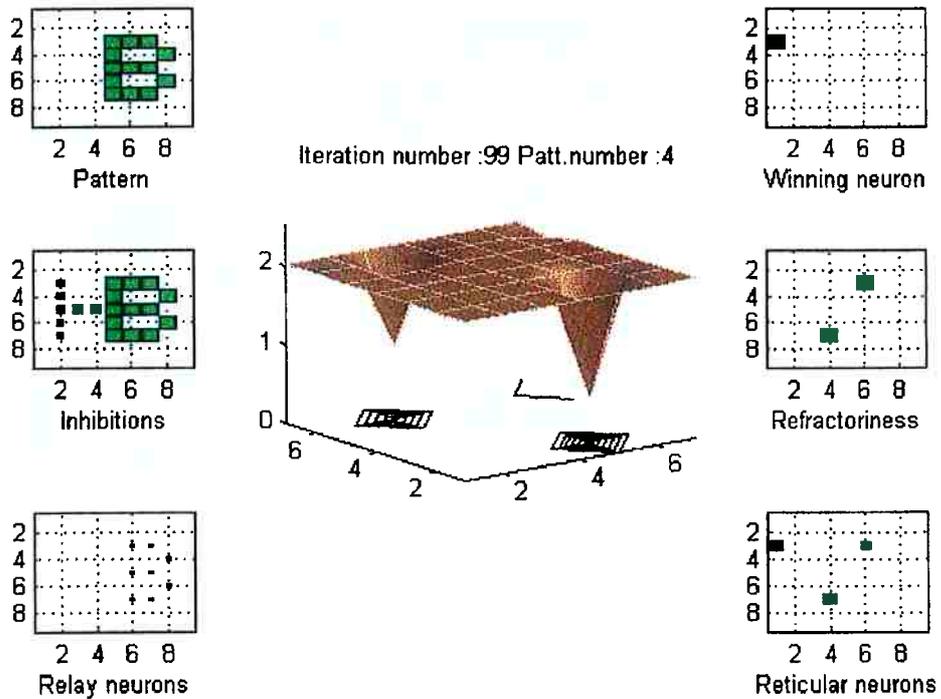


Figura 27

Nesta Figura 28 o potencial de membrana da maior parte dos 81 neurônios alcançou seu nível mais alto. Desde este momento e, ao longo das figuras que seguem até a Figura 31 este potencial estará descendo. Temos que indicar que tanto o aumento do potencial de membrana como a diminuição deste potencial acontece seguindo uma lei exponencial como é o caso também dos neurônios reais.

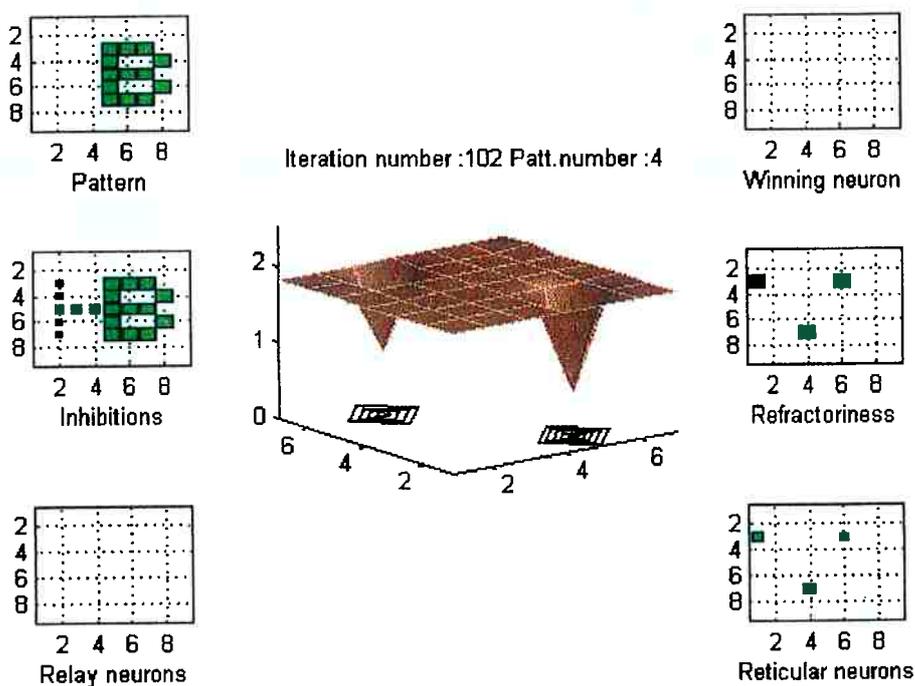
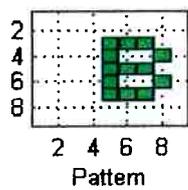
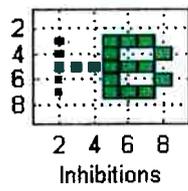


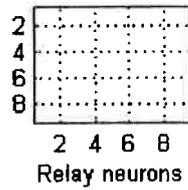
Figura 28



Pattern

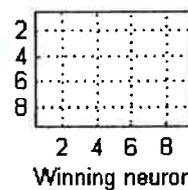
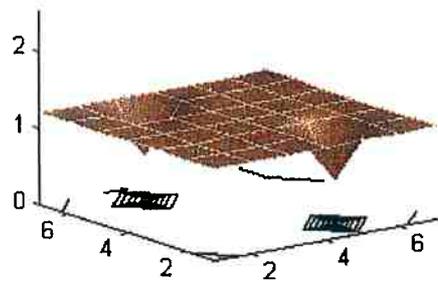


Inhibitions

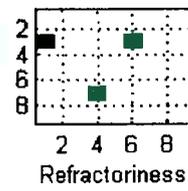


Relay neurons

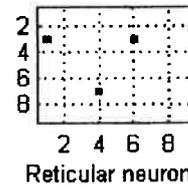
Iteration number :113 Patt.number :4



Winning neuron



Refractoriness



Reticular neurons

Figura 29

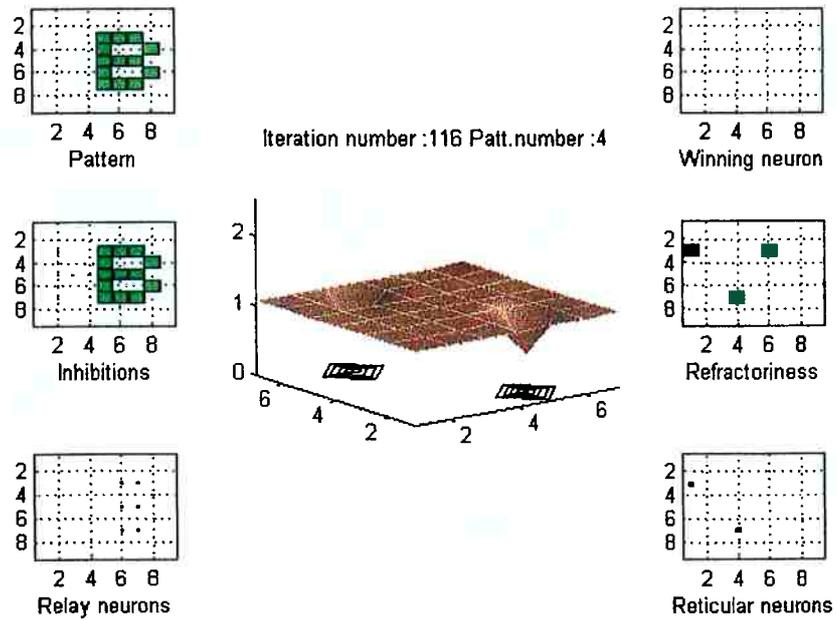


Figura 30

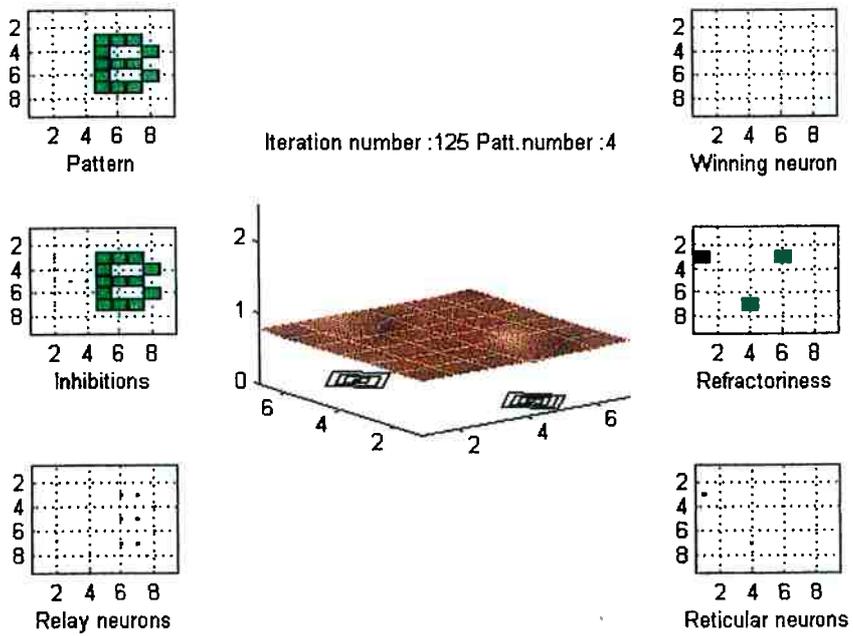


Figura 31

Na Figura 32 vemos a evolução do potencial dos 81 neurônios da camada reticular ao longo das 40 iterações durante as quais a letra "B" foi processada. A evolução dos neurônios ganhadores é facilmente identificável nesta figura.

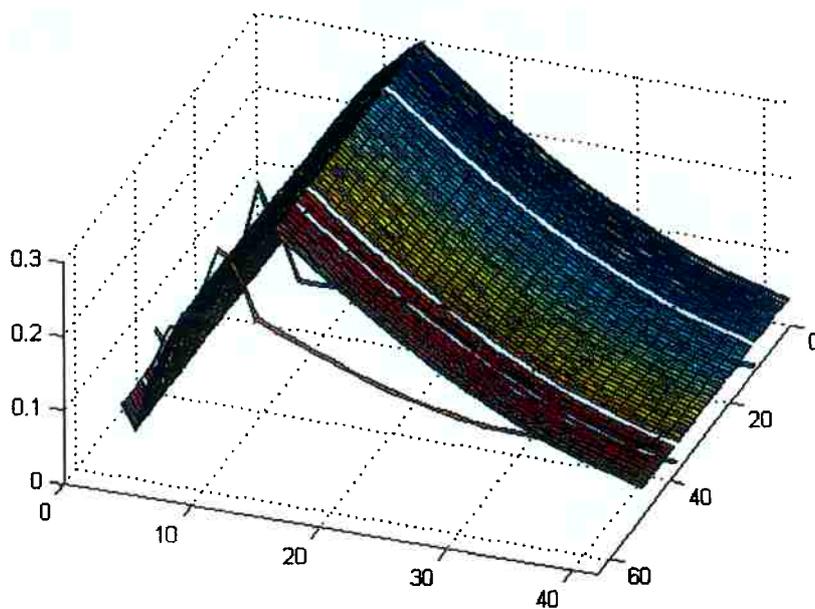


Figura 32

## ***DEMONSTRAÇÃO DA CAPACIDADE DE RECONSTRUÇÃO DE PADRÕES DA REDE TALÂMICA***

Um vez treinada a rede com os padrões: 4, H, B, 3, b, E e tendo sido apresentado cada padrão 4 vezes para realizar o treinamento (em cada apresentação a rede utiliza 40 iterações), foi testada a capacidade da rede para reconstruir um padrão incompleto. Neste momento a parte inferior da letra “H” é apresentada. O processo pelo qual a letra “H” vai se reconstruindo é apresentado da a Figura 33 até a Figura 37. Na Figura 33 pode ser visto que o neurônio reticular (5,8) foi o neurônio ganhador (o de maior ativação). Este neurônio inibe as entradas com uma inibição em forma de “E”. Esta primeira inibição seria a média dos padrões apresentados. Na verdade uma função do tipo sigmoide é aplicada às inibições com a finalidade de aumentar o contraste antes destas inibições serem mostradas na tela. Por isto, esta primeira inibição na realidade é da forma aproximada de “HE” que é a média de todos os caracteres alfanuméricos apresentados.

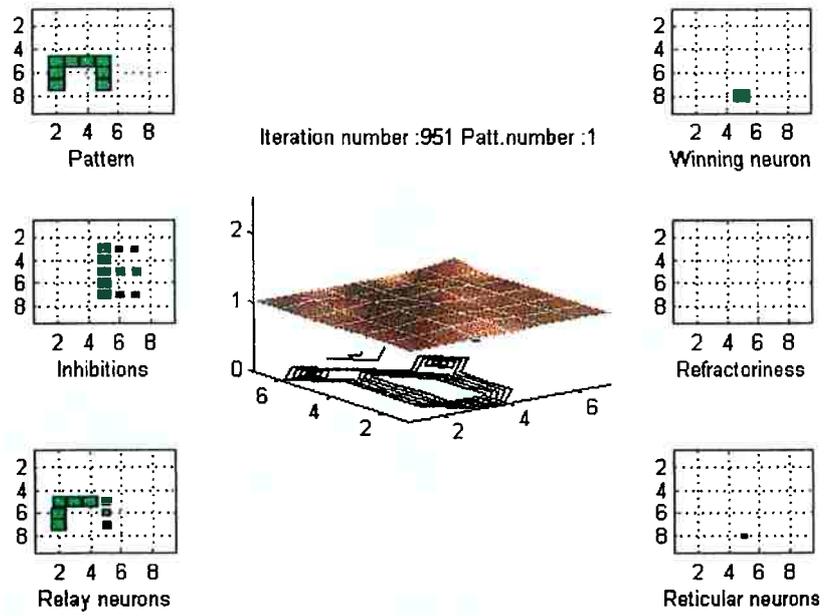


Figura 33

Na Figura 34 o neurônio reticular (5,8) entra em período refratário e aparece um segundo neurônio ganhador na camada reticular, o neurônio (8,8). A inibição criada por este neurônio é superposta como se fosse uma “máscara” à inibição do (5,8) criando o padrão de inibições da figura. Ainda faltam mais “mascaras” para serem superpostas sobre estas inibições. A inibição final acontecerá na hora que a maior parte dos neurônios (com a exceção dos neurônios que já dispararam) estejam com seu potencial de membrana num valor máximo, ou seja quando a superfície representada na figura esteja no seu nível mais alto.

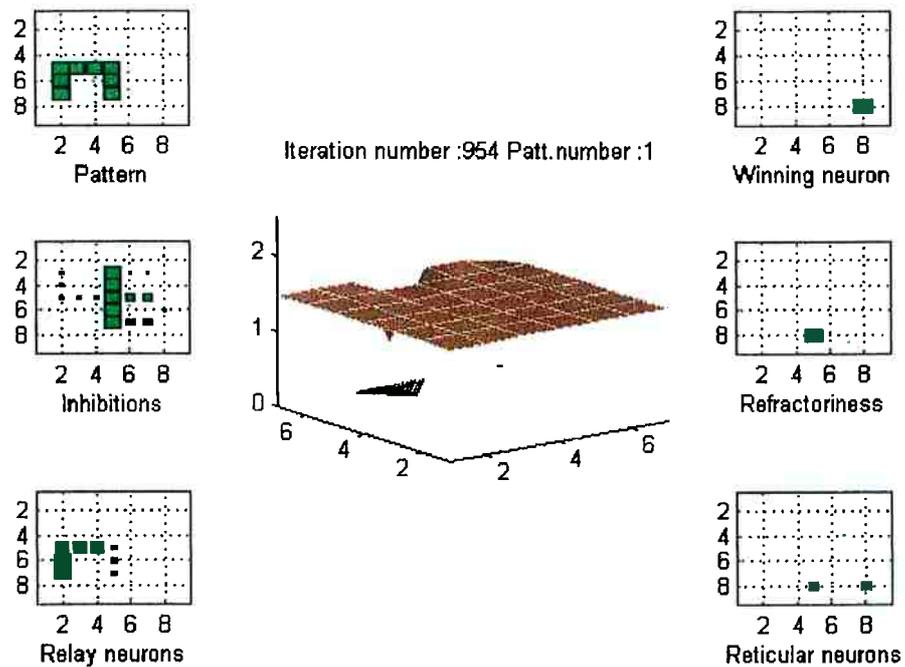


Figura 34

Na Figura 35 continua o processo de superposição das “máscaras” inibitórias que vão aos poucos recriando o padrão. Neste momento o neurônio (4,1) disparou e sua inibição somada às anteriores reconstruiu totalmente a letra “H”.

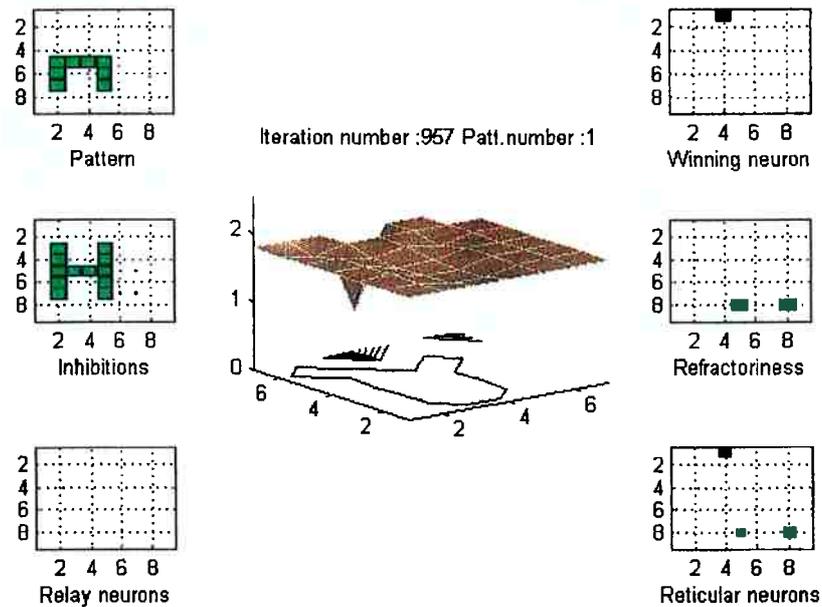
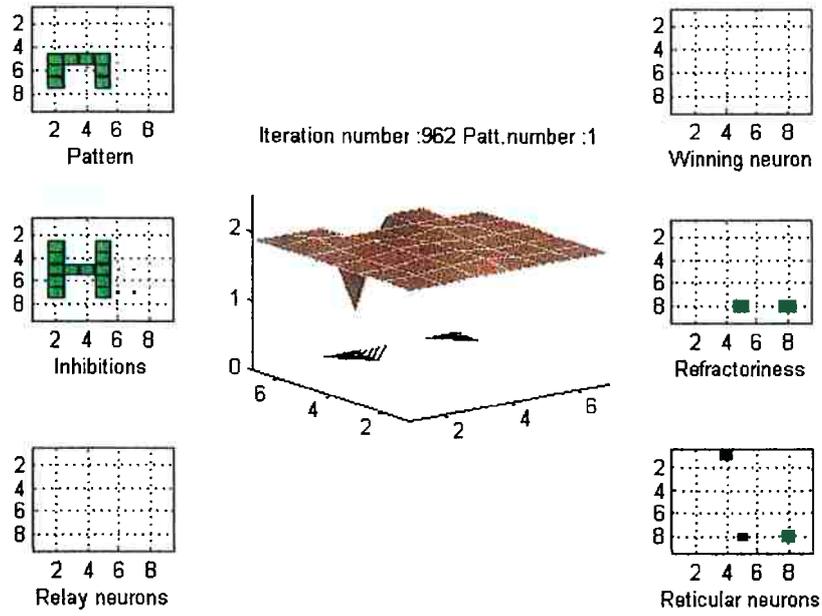
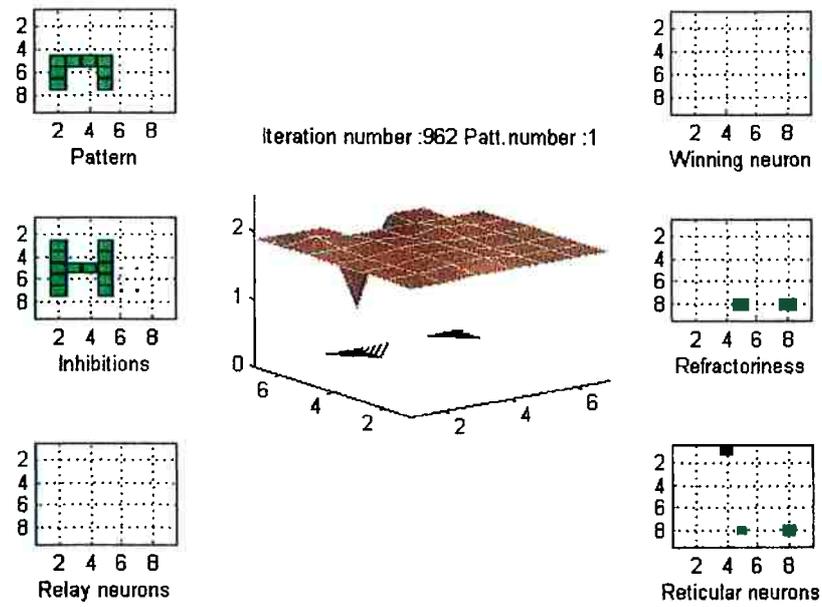


Figura 35

Nas seguintes figuras já não se superpõem mais “máscaras” inibitórias. O potencial base de membrana de todos os neurônios está descendo de forma síncrona até alcançar um potencial mínimo.



**Figura 36**



**Figura 37**



## ***CORRELAÇÃO COM A NEUROBIOLOGIA. IMAGENS ESTABILIZADAS NA RETINA.***

A capacidade da rede talâmica de completar padrões pode explicar o fenômeno que acontece quando uma imagem visual é estabilizada na retina. As referências 31 e 33 fornecem detalhes muito mais precisos em relação a este fenômeno. Resumindo, comentaremos que, quando uma imagem visual é fixada totalmente na retina, a maior parte dos bastonetes e cones da retina que processam a imagem, terminam saturando, e na verdade o olho termina não enviando a imagem inteira para o cérebro pelo nervo óptico. A única informação que é enviada em direção ao tálamo (o tálamo visual é chamado Núcleo Lateral Geniculado) são fragmentos da imagem que correspondem aos cones ou bastonetes que não estão saturados. Com esses fragmentos de informação o tálamo completa uma imagem que não necessariamente é a imagem que está sendo mostrada para o olho. Desta forma, a imagem real é substituída por uma alucinação visual que é o que realmente parece ver o sujeito que se submeteu ao experimento.

Os fragmentos de informação que são enviados ao tálamo tem uma vida curta porque os bastonetes e cones que enviam a informação terminam se saturando. Então outros bastonetes e cones saem da saturação e enviam uma nova mensagem em direção ao tálamo. Portanto a informação que chega ao tálamo é uma informação fragmentada que muda de uma hora para outra. O tálamo completa esta informação e, dependendo dela vai gerar um tipo de alucinação visual diferente.

Nas seguintes figuras o padrão “HB” (em verde) é apresentado para a rede. Dentro deste padrão aparecem os neurônios os quais estão sendo realmente ativos em vermelho. Com a escassa informação fornecida por estes neurônios, a rede talâmica vai reconstruir diferentes padrões dependendo dos neurônios ativos. Nas figuras 38 à 45 pode se observar o modo em que diferentes padrões foram reconstruídos a partir dos neurônios de entrada ativos (em vermelho). Observe-se que o resultado não é só um padrão inteiro recuperado como uma letra “E”, ou um “4”, etc. mas também pode ser um padrão misto de dois como na situação da Figura 43 no qual um padrão “4b” é reconstruído. Em outras ocasiões pode acontecer que só um fragmento dos padrões seja apresentado como na situação da Figura 45 no qual aparece uma “I” que, na verdade, não foi fornecida como padrão durante o treinamento, mas é na verdade uma parte essencial da maioria dos padrões apresentados. As figuras seguintes mostram os resultados e não requerem uma maior explicação.

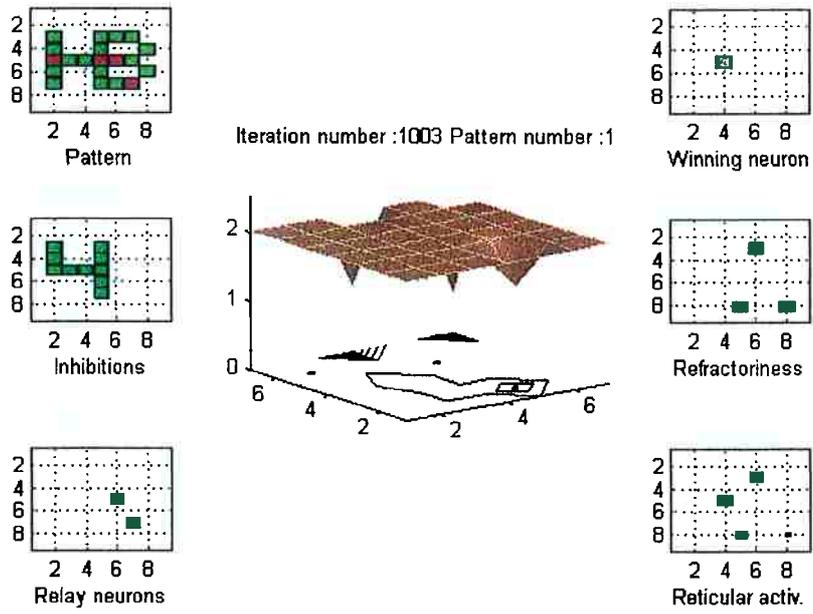


Figura 38

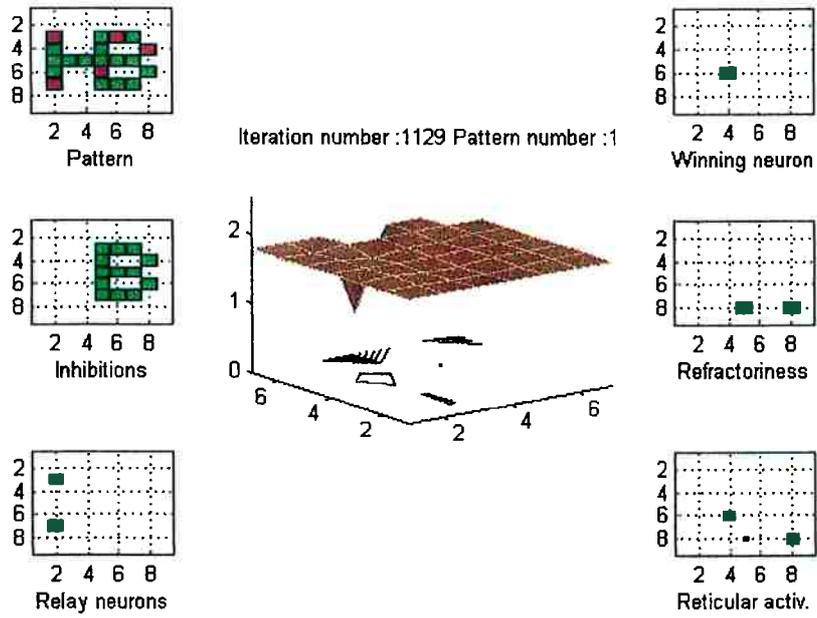
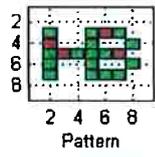


Figura 39



Iteration number :1176 Pattern number :1

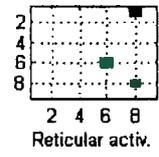
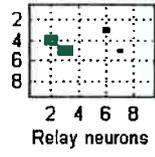
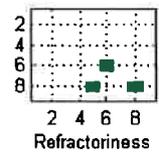
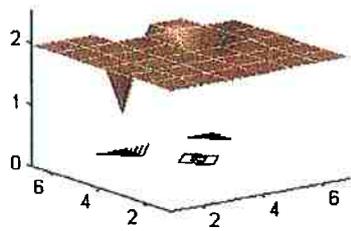
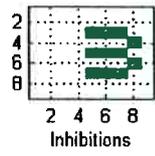
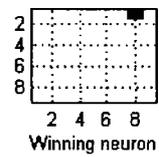
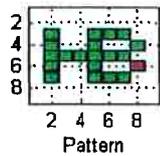


Figura 40



Iteration number :1219 Pattern number :1

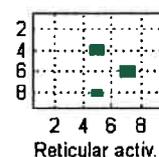
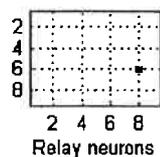
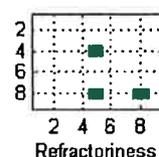
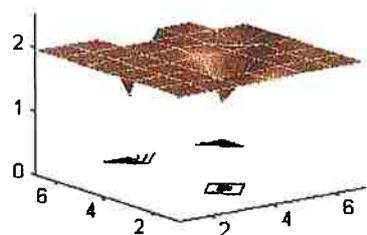
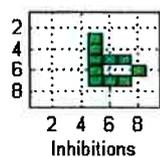
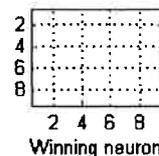


Figura 41

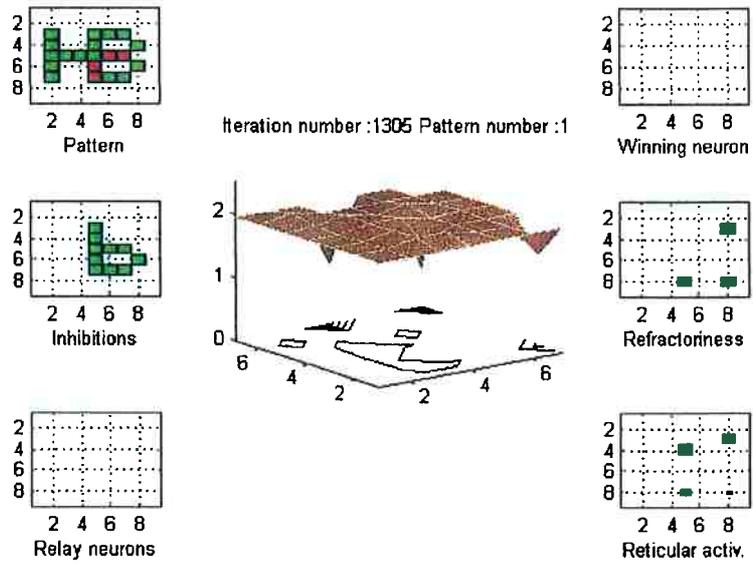


Figura 42

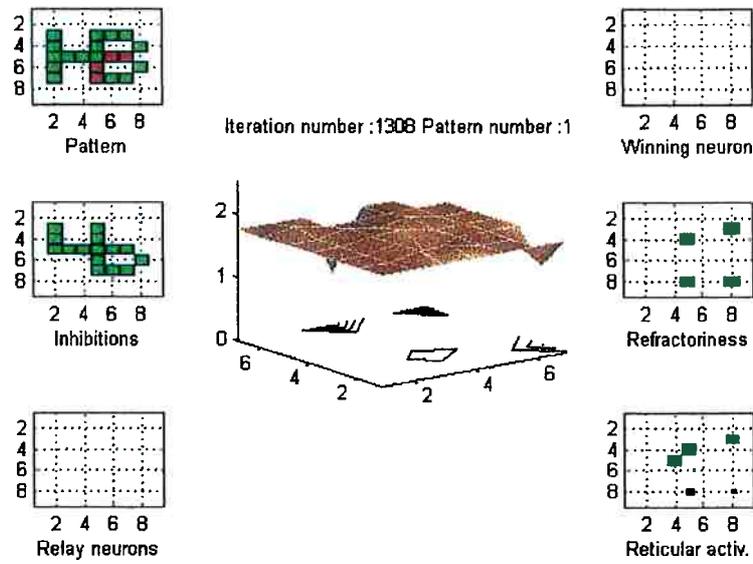


Figura 43

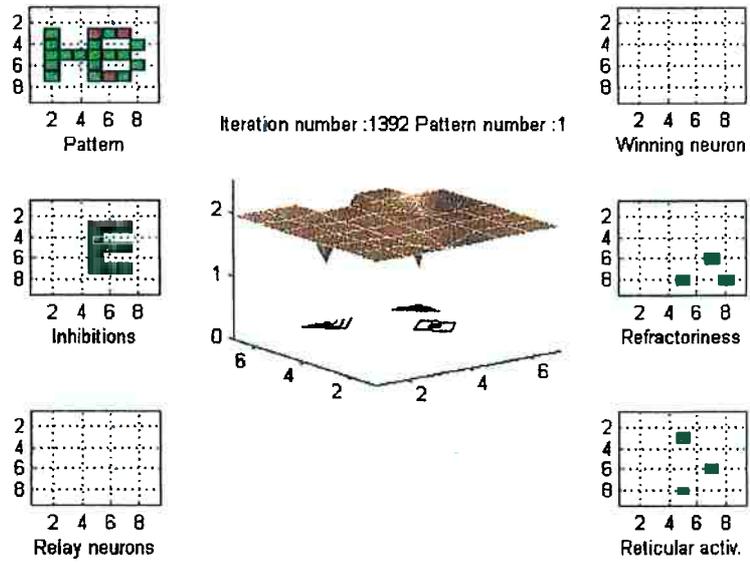


Figura 44

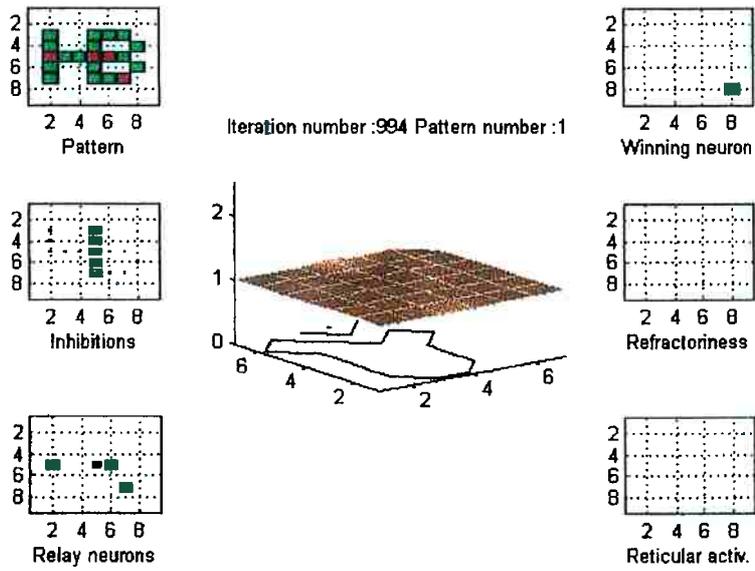


Figura 45

## ***EXPLICAÇÃO DETALHADA DE CADA MÓDULO DO PROGRAMA:***

### ▪ *Programa: PCAsinc\_osc1*

Este é o programa principal da rede talâmica a partir do qual são feitas chamadas às sub-rotinas dependentes dele. Como qualquer outro programa de computador este programa tem uma parte dedicada à inicialização das variáveis do programa que são basicamente:

a/ As atividades de cada um dos neurônios das duas camadas.

b/ O valor da matriz dos neurônios em refratário que é uma matriz que está preenchida inicialmente de números 1. Os zeros na matriz representam os neurônios em período refratário.

c/ Os pesos entre as duas camadas  $w_{tr}$  e  $w_{rt}$  são ajustados pela função  $\text{mod\_pesos}$  e inicialmente são de um valor pequeno.

d/ O parâmetro concentração de neuro-transmissor permite ajustar a intensidade do aumento ou diminuição do reforço sináptico em cada iteração do algoritmo.

A atividade de cada neurônio reticular está modelada pela equação:

$$\text{activacion\_reticular\_base} = \text{activacion\_reticular\_base} + I_h \cdot \text{refractoriness} + I_n;$$

onde a ativação<sub>reticular\_base</sub> corresponde à soma das entradas multiplicadas pelos pesos sinápticos enquanto  $I_h$  e  $I_n$  são conductâncias iônicas específicas dos neurônios reticulares reais que seguem uma evolução exponencial atualizada em cada iteração:

$In(i,j)=-Ink(i,j)*(1-\exp(-activacion\_reticular(i,j)/0.14));$

Além disto existem conexões dendro-dendríticas entre os neurônios reticulares. isto permite que o efeito de uma das condutâncias iônicas seja comunicado parcialmente aos neurônios reticulares adjacentes. isto se consegue a través da rotina:

$In(i,j)=((14*peso\_entre\_dendroden)*In(i,j)+peso\_entre\_dendroden*(In(i+1,j)+In(i-1,j)+In(i,j-1)+In(i,j+1)));$

A distorção que acontece nos neurônios nas bordas da rede por causa da anterior equação é eliminada através do ajuste nas matrizes intermediárias B e C.

Os padrões sucessivos são apresentados à rede na hora que a média da ativação reticular dos neurônios se encontra na parte mais baixa das oscilações do ciclo de oscilação dos neurônios. A função para a captura dos padrões se chama `coge_el_patrón` (que em português seria `pega_o_patrão`).

Finalmente a última seção do programa principal é dedicada às saídas das figuras pela tela do computador mediante a função `gráficoPCA sincosc(...)`.

❖ Código:

```
limite_superior=0.3;  
limite_inferior=0;
```

```
peso_entre_dendroden=0.01;%El valor maximo aqui es 0.25. El valor habitual es 0.001  
concentracion_neurotransmisor=50;%Base 150. Podriamos decir que este numero da el numero de
```

```

moleculas de neurotransmisor en una vesicula sinaptica
latencia_inhibicion=0.94;%%Valor que varia entre 0 y 1. Parece que un valor de 0.92 es demasiado pequeno.
latencia_refract=0.95;
inc_potencial_ganadora=0.08*limite_superior;%%El valor base es 0.15*limite_superior
%%inc_potencial_ganadora=1;

random_presentation=0;%%Permite que los patrones sean seleccionados de forma aleatoria o sucesivamente

activacion_reticular=0.0.*rand(filas_ret,columnas_ret)+0.0.*ones(filas_ret ,columnas_ret);

Ink=zeros(size(activacion_reticular));
Ihk=zeros(size(activacion_reticular));
Estimulo=zeros(size(activacion_reticular));
activacion_reticular_base=zeros(size(activacion_reticular));
pendiente=zeros(size(activacion_reticular));
activacion_reticular=0.001.*ones(size(activacion_reticular));
activacion_reticular_inputs=0.001.*ones(size(activacion_reticular));

activacion_ret_old=0.5.*activacion_reticular;

Ih=zeros(filas_ret, columnas_ret);
In=zeros(filas_ret, columnas_ret);

Media=zeros(filas_ret,columnas_ret);
preparado_patron=4;

%%INICIALIZACION PARAMETROS SUBROUTINAS

%%PARAMETROS DE: ESTIM_RETIC.
prob_rele=zeros(filas_rel, columnas_rel);

%%PARAMETROS DE: PROB_RELE.
ret_in_order=zeros(filas_ret, columnas_ret);
matriz_inhibiciones=zeros(filas_rel,columnas_rel);

%%PARAMETROS DE: PROB_MAX_RET
end_flag=1;

activacion_reticular=0.01.*rand(filas_ret,columnas_ret);
activacion_ret_old=ones(filas_ret,columnas_ret);%%Atencion, este valor debe ser suficientemente grande para
que se pueda encontrar el primer prob_max_ret
prob_max_ret(filas_ret, columnas_ret)=1;
refractoriness=ones(filas_ret,columnas_ret);
neurona_ganadora=zeros(filas_ret, columnas_ret);
incremento_old=ones(filas_ret, columnas_ret);
new_refract=zeros(filas_ret, columnas_ret);

Init_wrtn=1;
Init_wrttd=50;

```

```
Init_wtrn=0.05; %Es importante que este valor inicial sea pequeño porque así al principio no queda ninguna
neurona sin activar los canales Ih.
```

```
Init_wtrd=50;
```

```
m=filas_ret*columnas_ret;
```

```
wtrn=Init_wtrn*ones(size(P,1)*size(P,2), m);
wtrd=Init_wtrd.*ones(size(P,1)*size(P,2),m);
```

```
wtrn=Init_wtrn.*rand(size(P,1)*size(P,2),m);
wtrd=Init_wtrd.*ones(size(P,1)*size(P,2),m);
```

```
wtr=transpose(wtrn./wtrd);
wrt=(wtrn./wtrd);
```

```
wtr_ini=wtr;
wrt_ini=wrt;
```

```
end %end if (cont==0)
```

```
for iter=1:iteraciones
    close;
end;
```

```
show_graph=input('Escriba 1 para mostrar grafico normal y 0 para no mostrar ');
train_test=input('Escriba 0 para patrones normales y 1 para patrones parciales ');
graf_fin=input('Escriba 1 para mostrar grafico final y 0 para no mostrar ');
```

```
if (train_test==1) np=1;
end
```

```
for iter=1:iteraciones
    numero=0;
    it=it+1;
    if (mean(mean(activacion_reticular))>=0.5*limite_superior)
        preparado_patron=0;
    end
```

```
    if (mean(mean(activacion_reticular))<0.2*limite_superior)
        preparado_patron=preparado_patron+1;
    end
```

```
if (preparado_patron==8);
```

```
    if (train_test==0)
        [P,np]=coge_el_patron(random_presentation,np);
    else
        [P,np]=coge_el_patron_test(random_presentation,np);
    end
```

```

% TODAS LAS SIGUIENTES INICIALIZACIONES PUEDEN BORRARSE PARA UN
FUNCIONAMIENTO MAS LIBRE
refractoriness=ones(filas_ret,columnas_ret);
ret_in_order=zeros(filas_ret,columnas_ret);
matriz_inhibiciones=zeros(filas_rel,columnas_rel);
activacion_reticular=zeros(filas_ret, columnas_ret);
activacion_reticular_base=zeros(filas_ret, columnas_ret);
activacion_reticular_inputs=zeros(filas_ret, columnas_ret);
Ih=zeros(filas_ret,columnas_ret);
Ihk=0.2*limite_superior.*ones(filas_ret,columnas_ret);
In=zeros(filas_ret,columnas_ret);
Ink=zeros(filas_ret,columnas_ret);
activacion_ret_old=0.0.*activacion_reticular;
new_refract=zeros(filas_ret, columnas_ret);
prob_max_ret=zeros(filas_ret, columnas_ret);
B=zeros(filas_ret, columnas_ret);
C=zeros(filas_ret, columnas_ret);
end
if (rem(iter,ceil(iteraciones/200))==0)
it
end
if (it>1)
[prob_rele, matriz_inhibiciones]=
prob_rele_f(ret_in_order,wrt,P,matriz_inhibiciones,0.7);
activacion_reticular_inputs=
estim_retic_f(prob_rele,wtr,filas_ret,columnas_ret);%activacion_reticular es la matriz de la iteracion anterior
[prob_max_ret,ret_in_order,activacion_ret_old,refractoriness,incremento, new_refract]=
prob_max_ret_f(activacion_reticular, activacion_reticular_inputs, activacion_ret_old, prob_max_ret,
ret_in_order,latencia_inhibicion, refractoriness,incremento_old, limite_superior, new_refract,
latencia_refract);
prob_max_ret;
if(max2D(prob_max_ret)>0|(train_test==1))
[wtrn,wtrd,wtrn,wtrd]=mod_pesos_f(wtrn,wtrd,wtrn,wtrd,prob_rele,prob_max_ret,P,matriz_inhibiciones,co
ncentracion_neurotransmisor);
wtr=transpose(wtrn./wtrd);
wrt=(wtrn./wtrd);
end
end
activacion_reticular_base=activacion_reticular_base+Ih.*refractoriness+In; %Es importante aplicar el
periodo refractario no solo a los parametros de disparo sino a la propia entrada y al Ih
activacion_reticular=activacion_reticular_inputs.*refractoriness+activacion_reticular_base+(inc_potencial_g
anadora).*(prob_max_ret>zeros(filas_ret, columnas_ret));
B=activacion_reticular;
[valor_prob_max_ret,x_ganadora, y_ganadora]=max2D(prob_max_ret);
Ihk(x_ganadora, y_ganadora)=0.0;
Ink(x_ganadora, y_ganadora)=0.05*limite_superior;

for i=2:filas_ret-1
for j=2:columnas_ret-1

if (activacion_reticular(i,j)>0.98*limite_superior)
Ihk(i,j)=0.0;
Ink(i,j)=0.05*limite_superior;

```

```

        end;
        if (activacion_reticular(i,j)<=(0.02+limite_inferior))
            Ihk(i,j)=0.2*limite_superior;
            Ink(i,j)=0.0;
        end;
In(i,j)=-Ink(i,j)*(1-exp(-activacion_reticular(i,j)/0.14));
Ih(i,j)=Ihk(i,j)*(exp(-activacion_reticular(i,j)/0.2));
In(i,j)=((1-4*peso_entre_dendroden)*In(i,j)+peso_entre_dendroden*(In(i+1,j)+In(i-1,j)+In(i,j-1)+In(i,j+1)));
%B(i,j)=((1-4*peso_entre_dendroden)*B(i,j)
+peso_entre_dendroden*(B(i+1,j)+B(i-1,j)+B(i,j-1)+B(i,j+1)));
end
end

C=B;
C(1,1)=(B(1,1)+B(1,2)+B(2,1))/3;
C(1,columnas_ret)=(B(1,columnas_ret-1)+B(2,columnas_ret)+B(1,columnas_ret))/3;
C(filas_ret,1)=(B(filas_ret-1,1)+B(filas_ret,2)+B(filas_ret,1))/3;
C(filas_ret,columnas_ret)=(B(filas_ret,columnas_ret)+B(filas_ret-1,columnas_ret)+B(filas_ret,columnas_ret-1))/3;

C(1,:)=B(2,:);
C(filas_ret,:)=B(filas_ret-1,:);
C(:,1)=B(:,2);
C(:,columnas_ret)=B(:,columnas_ret-1);
activacion_reticular=C;

    if(graf_fin==1)
        for i=2:filas_ret-1
            for j=2:columnas_ret-1
                numero=numero+1;
                rib_graph(numero, iter)=activacion_reticular(i,j);
            end
        end
    end %end graf_fin

    if (show_graph==1)
        pause;
        graficoPCAsincosc(matriz_inhibiciones,activacion_reticular, end_flag, it, 0.5, filas_rel,columnas_rel,P,
ret_in_order, prob_max_ret, prob_rele, refractoriness, np)
    end

end %end for iter

if (graf_fin==1)
figure
ribbon(rib_graph);
view(110,50)
axis([0,(filas_ret-1)*(columnas_ret-1),0, iter,0,limite_superior]);
wtr_end=wtr;
wrt_end=wrt;
end
rib_graph=[];

```

▪ *Programa: Coge\_el\_patron*

Este programa permite a captura dos padrões de treinamento da rede, bem seja em ordem aleatório (colocando a variável `random_presentation` igual a 1) ou em ordem seqüencial. Os padrões são as matrizes que aparecem no programa.

❖ Código:

```
function [P,np]=coge_el_patron(random_presentation,np)
```

```
numero_de_patrones=8;
```

```
if (np>=numero_de_patrones)  
    np=1;  
end
```

```
Pat(:,1)=[0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0;  
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0;  
0.0 1.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0;  
0.0 1.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0;  
0.0 1.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0;  
0.0 1.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0;  
0.0 1.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0;  
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0;  
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0];
```

```
Pat(:,2)=[0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0;  
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0;  
0.0 1.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0;  
0.0 1.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0;  
0.0 1.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0;  
0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0;  
0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0;  
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0;  
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0];
```

```
Pat(:,3)=[0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0;
```



```
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0;
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0];
```

```
Pat(:,8)=[0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0;
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0;
0.0 0.0 0.0 0.0 0.0 1.0 1.0 0.0 0.0;
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0;
0.0 0.0 0.0 0.0 0.0 1.0 1.0 0.0 0.0;
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0;
0.0 0.0 0.0 0.0 0.0 1.0 1.0 0.0 0.0;
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0;
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0];
```

```
if (random_presentation==1)
    P=Pat(:,ceil(numero_de_patrones*rand));
else
    P=Pat(:,np);
    np=np+1;
end
```

- *Programa: Prob\_max\_ret\_f*

Este programa permite dizer em que momento vão disparar cada um dos neurônios reticulares. Assim estabelece algumas regras para a alternância dos neurônios como por exemplo: só serão disparados os neurônios cujo incremento de ativação ultrapasse um determinado limiar e somente serão disparados na hora em que o neurônio vencedor que estava disparando tenha um potencial decrescente de determinado valor. Dentro desta rotina aparece a rotina max2D que permite calcular o máximo de uma matriz bidimensional (Este tipo de operação simples surpreendentemente não existe no Matlab) cujo valor é armazenado na matriz prob\_max\_ret. Também nesta importante rotina são calculados os neurônios que passam a período refratário, e também a matriz “ret\_in\_order” que pondera os neurônios reticulares que foram disparados para conseguir uma inibição latente que permanece após os neurônios entrarem no período refratário.

- **Código:**

```
function[prob_max_ret, ret_in_order, activacion_ret_old, refractoriness,incremento_old,
new_refract]=prob_max_ret_f(activacion_reticular, activacion_reticular_inputs, activacion_ret_old,
prob_max_ret, ret_in_order,latencia_inhibicion, refractoriness, incremento_old, limite_superior,
new_refract, latencia_refract)
[n_filas,n_columnas]=size(activacion_reticular);
ceros=zeros(n_filas,n_columnas);
unos=ones(n_filas,n_columnas);
activacion_reticular;
activacion_ret_old;

[valor_prob, x_old, y_old]=max2D(prob_max_ret>ceros);

incremento=activacion_reticular-activacion_ret_old;

if (valor_prob~=0)
    if (incremento(x_old,y_old)<-0.01)% Base -0.025. Para stabilized images fue cambiado a -0.0075
        new_refract=new_refract+ceros;
```

```

new_refract(x_old,y_old)=1;
prob_max_ret=ceros;
end %if (incremento(x_old,y_old)<-0.01)

else %if (valor_prob~=0)
    incremento_positivo=(incremento>0.01*0.3.*unos);%0.3 es el limite superior y este valor es un ajuste
entre el necesario para que el potencial despegue como para que no vuelva a subir cuando aterrice basado en
el Ihk
    prob_max_ret=ceros;
    [valor,x,y]=max2D((incremento_positivo.*refractoriness).*activacion_reticular);

    if (valor~=0)
        prob_max_ret(x,y)=(activacion_reticular(x,y)/limite_superior);
    else
        prob_max_ret(x,y)=0;
    end

    %prob_max_ret(x,y)=sin(atan(valor));
    %prob_max_ret=prob_max_ret.*refractoriness;%refractoriness es una matriz con ceros en
    %aquellos elementos que fueron ganadores en ocasiones anteriores
    ret_in_order=(latencia_inhibicion.*ret_in_order).*(latencia_inhibicion.*ret_in_order>0.1.*unos);

    %refractoriness=((ret_in_order<=ceros)|(refractoriness)).*(ret_in_order<0.12.*unos);
    ret_in_order=prob_max_ret+ret_in_order;
    new_refract=((new_refract.*latencia_refract>0.05.*unos)).*new_refract.*latencia_refract;%Para un
funcionamiento libre el valor limite es 0.2 en vez de 0.05
    refractoriness=~(new_refract>ceros);
end %if (valor_prob~=0)
activacion_ret_old=activacion_reticular;
incremento_old=incremento;

```

▪ *Programa: Prob\_rele\_f*

Esta rotina calcula a ativação dos neurônios da primeira camada que é a soma do padrão de entrada menos as inibições de feed-back que vem desde os neurônios da segunda camada. Para conseguir avaliar estas inibições é necessário multiplicar uma matriz bidimensional (ret\_in\_order de 9x9) pelos pesos que é uma outra matriz bidimensional (wrt de 81x81). Isto é impossível de ser feito diretamente e por isto a matriz ret\_in\_order é transformada numa matriz (81,1) mediante a função do matlab “reshape”.

❖ Código:

```
function[prob_rele, matriz_inhibiciones]=prob_rele_f(ret_in_order,wrt,B,matriz_inhibiciones,inhib_param)
%Aqui el B es una columna a diferencia de ret_in_order y prob_rele que son matrices

[filas_ret, columnas_ret]=size(ret_in_order);
[filas_B,columnas_B]=size(B);
ret_shaped=reshape(ret_in_order,filas_ret*columnas_ret,1);
ret_shaped=ret_shaped./(max(ret_shaped)+0.0000001);
Bshaped=reshape(B,filas_B*columnas_B,1);
inhibicion_fila=(wrt*ret_shaped);
prob_rele_col=(Bshaped-inhibicion_fila+0.00.*rand(filas_B*columnas_B,1)).*((Bshaped-
inhibicion_fila+0.0.*rand(filas_B*columnas_B,1))>zeros(filas_B*columnas_B,1));
prob_rele=reshape(prob_rele_col,filas_B,columnas_B);
matriz_inhibiciones=reshape(inhibicion_fila, filas_B, columnas_B);
```

- *Programa: Stim\_retic\_f*

Este programa calcula a ativação de cada um dos neurônios reticulares multiplicando-se as atividades dos neurônios “relay” pelos pesos de cada neurônio reticular. O problema da falta de concordância das dimensões das matrizes a serem multiplicadas é resolvido mediante a função do Matlab “reshape” de forma análoga ao caso da rotina já apresentada prob\_rele\_f.

❖ Código:

```
function[estim_retic]=estim_retic_f(prob_rele,wtr,filas_ret,columnas_ret)
[filas,columnas]=size(prob_rele);
prob_rele_column=reshape(prob_rele,filas*columnas,1);
[filasw,columnasw]=size(wtr);
estim_retic=reshape(((wtr)*prob_rele_column)./columnasw,filas_ret,columnas_ret);
```

- *Programa: mod\_pesos\_f*

Esta matriz permite preparar os padrões que mais tarde vão entrar na função “calculate\_weights” para calcular os pesos entre os neurônios da camada de entrada e a camada de saída. As matrizes de pesos: wtr e wrt têm uma dimensão de 81x81, enquanto as matrizes de entrada e saída prob\_rele, prob\_max\_ret e P são matrizes de 9x9. Para conseguir utilizar estas matrizes de entrada e saída visando alterar os pesos sinápticos é preciso reformatá-las mediante a função “reshape” que já foi apresentada na explicação das funções anteriores.

❖ Código:

```
function[wtrn,wtrd,wtrn,wtrd]=mod_pesos_f(wtrn,wtrd,wtrn,wtrd,prob_rele,prob_max_ret,P,matriz_inhibiciones,iter)
[n_filas_rel, n_columnas_rel]=size(prob_rele);
[n_filas_ret, n_columnas_ret]=size(prob_max_ret);
prob_rele1=reshape(prob_rele,n_filas_rel*n_columnas_rel,1);
prob_max_ret1=reshape(prob_max_ret,n_filas_ret*n_columnas_ret,1);
inhib=reshape(matriz_inhibiciones,n_filas_rel*n_columnas_rel,1);
P1=reshape(P,n_filas_rel*n_columnas_rel,1);
for (i=1:iter)
    rel_disp=linear(prob_rele1);%En el caso de colocar una funcion sigmoidal utilizar la funcion "sigmoid1"
    rel_disp=logsig((prob_rele1-0.03)*15);
    ret_disp=linear(prob_max_ret1)+0.00.*rand(n_filas_ret*n_columnas_ret,1);

    pat_disp=linear(P1);
    [wtrn, wtrd]=calculate_weights(ret_disp,rel_disp,wtrn,wtrd);
    [wtrn, wrtd]=calculate_weights(ret_disp,pat_disp,wtrn,wtrd);
end
```

- *Programa: calculate\_weights*

Este programa permite atualizar o numerador e denominador das probabilidades condicionais que são os pesos entre os neurônios.

- ❖ Código:

```
function[matrix_num, matrix_den]=calculate_weights(input, output, matrix_num, matrix_den)
matrix_num=output*input'+matrix_num;
matrix_den=ones(size(output))*input'+matrix_den;
```

- *Programa: GraficoPCAsincosc*

Este programa fornece a saída dos resultados de cada iteração na tela do computador. Ele faz um uso extensivo da função “subplot” para situar cada um dos gráficos na posição adequada na tela do computador.

❖ Código:

```
function graficoPCAsincosc(matriz_inhibiciones,A, end_flag, iter, factor_elevacion,filas_rel,columnas_rel,P,
ret_in_order, prob_max_ret, prob_rele,refractoriness, np)

D=A;
matriz_inh_graf= matriz_inhibiciones;
[filas_ret, columnas_ret]=size(A);
elevacion=factor_elevacion.*ones(filas_ret-2, columnas_ret-2);
D(:,1)=[];
matriz_inh_graf(:,1)=[];
D(1,:)=[];
matriz_inh_graf(1,:)=[];
D(filas_ret-1,:)=[];
matriz_inh_graf(filas_ret-1,:)=[];
D(:,columnas_ret-1)=[];
matriz_inh_graf(:,columnas_ret-1)=[];

pause(0.0000000001);
subplot('position', [0.30,0.30,0.4, 0.4])
colormap(copper);
mesh(5.*D+elevacion);
hold on
surf(5.*D+elevacion), shading interp;
hold on;
media=mean(mean(matriz_inhibiciones));
contour(D)
title(strcat('Iteration number ', num2str(iter), ' Part.number ', num2str(np)));

grid off;
hold off;
axis([1,filas_ret-2,1,columnas_ret-2,0,2.5]);
view(-37.5, 20);

subplot('position', [0.05,0.77,0.16, 0.16])
grafico_red(P);
xlabel('Pattern');

subplot('position', [0.05,0.45,0.16, 0.16])%El 42 inicial fue cambiado a 47
grafico_red(logsig(15.*(matriz_inhibiciones-0.6.*ones(size(matriz_inhibiciones))))));
xlabel('Inhibitions');
```

```
subplot('position', [0.77,0.45,0.16, 0.16])
grafico_red(3000.*(~refractoriness));
xlabel('Refractoriness');
```

```
subplot('position', [0.05,0.10,0.16, 0.16])
grafico_red(prob_rele,1,0);
xlabel('Relay neurons');
```

```
subplot('position', [0.77,0.77,0.16, 0.16])
grafico_red(2000.*prob_max_ret);
xlabel('Winning neuron');
```

```
subplot('position', [0.77,0.1,0.16, 0.16])
grafico_red(ret_in_order);
xlabel('Reticular neurons');
```

- Programa: *grafico\_red*.

Este programa não é mais que o programa “hintonw” que vem com as “neural networks’ Pctools” do Matlab e portanto é um programa que já estava pronto e não foi realizado pelo aluno. Unicamente o nome foi mudado porque foram cogitadas algumas mudanças no programa que, finalmente, não precisaram ser realizadas.

❖ Código

```
function grafico_red(w,max_m,min_m)
if nargin < 1,error('Not enough input arguments.');
```

```
end
if nargin < 2, max_m = max(max(abs(w))); end
```

```
if nargin < 3, min_m = max_m / 100; end
```

```
if max_m == min_m, max_m = 1; min_m = 0; end
```

```
xn1 = [-1 -1 +1]*0.5;
```

```
xn2 = [+1 +1 -1]*0.5;
```

```
yn1 = [+1 -1 -1]*0.5;
```

```
yn2 = [-1 +1 +1]*0.5;
```

```
xn = [-1 -1 +1 +1 -1]*0.5;
```

```
yn = [-1 +1 +1 -1 -1]*0.5;
```

```
xp = [xn [-1 +1 +1 +0 +0]*0.5];
```

```
yp = [yn [+0 +0 +1 +1 -1]*0.5];
```

```
[S,R] = size(w);
```

```
cla reset
```

```
hold on
```

```
set(gca,'xlim',[0 R]+0.5);
```

```
set(gca,'ylim',[0 S]+0.5);
```

```
set(gca,'xlimmode','manual');
```

```
set(gca,'ylimmode','manual');
```

```
xticks = get(gca,'xtick');
```

```
set(gca,'xtick',xticks(find(xticks == floor(xticks))))
```

```
yticks = get(gca,'ytick');
```

```
set(gca,'ytick',yticks(find(yticks == floor(yticks))))
```

```

set(gca,'ydir','reverse');
if get(0,'screendepth') > 1
    set(gca,'color',[1 1 1]*.5);
    set(gcf,'color',[1 1 1]*.3);
end

for i=1:S
    for j=1:R
        m = sqrt((abs(w(i,j))-min_m)/max_m);
        m = min(m,max_m)*0.95;
        if real(m)
            if w(i,j) >= 0
                fill(xn*m+j,yn*m+i,[0 0.8 0])
                plot(xn1*m+j,yn1*m+i,'w',xn2*m+j,yn2*m+i,'k')
            elseif w(i,j) < 0
                fill(xn*m+j,yn*m+i,[0.8 0 0]);
                plot(xn1*m+j,yn1*m+i,'k',xn2*m+j,yn2*m+i,'w');
            end
        end
    end
end

plot([0 R R 0 0]+0.5,[0 0 S S 0]+0.5,'w');

grid on

```

- *Programa: max2D*

Este programa calcula o valor e as coordenadas x e y do elemento de valor máximo numa matriz bidimensional. A rotina “max” do Matlab realiza esta mesma tarefa em matrizes unidimensionais.

❖ Código

```
function[valor,fil,columna]=max2D(a)
[max_por_col,fil_del_maximo]=max(a);
[valor,columna]=max(max_por_col);
[valor,fil]=max(a(:,columna));
```

- *Programa: sigmoid1*

Este programa fornece um disparo binário cuja probabilidade resulta da aplicação da função sigmoide sobre a variável `act_vect`.

❖ Código

```
function[bin_vect]=sigmoid1(act_vect)
sigm=logsig((act_vect-0.2)*15);

rand_vect=rand(size(act_vect));
bin_vect=(sigm>=rand_vect);
```

- *Programa: linear*

Este programa fornece um disparo binário cuja probabilidade resulta da aplicação de uma função linear sobre a variável `act_vect`.

❖ Código

```
function[bin_vect]=linear(act_vect)
lin=act_vect;
rand_vect=rand(size(act_vect));
bin_vect=(lin>=rand_vect);
```

- *Programa: Muestra\_pesos*

Este programa mostra em seqüência 81 janelas com os 81 pesos de entrada dos 81 neurônios relay.

❖ Código

```
function muestra_pesos(wrt, filas_ret, columnas_ret)
for i=1:filas_ret*columnas_ret

    hintonw(reshape(wrt(:,i),filas_ret,columnas_ret));
    pause(0.000001);
end
```

- *Programa: Muestra\_pesos\_tr*

Este programa muestra en seqüência 81 janelas com os 81 pesos de entrada dos 81 neurônios reticulares.

❖ Código

```
function muestra_pesos_tr(wtr, filas_ret, columnas_ret)
for i=1:filas_ret*columnas_ret
    hintonw(reshape(wtr(i,:),filas_ret,columnas_ret));%Esta funcion fue ajustada asi para ver
    % los pesos que corresponderian a las probabilidades P(r/y) donde r corresponde a la activacion de una
    neurona rele
    pause(0.01);
end
```

- *Programa: stabilized\_images*

Este programa é equivalente ao programa principal PCAsync\_osc1, porém adaptado para o modelo das imagens estabilizadas na retina. Basicamente cada vez que pega um padrão seleciona aleatoriamente alguns dos “pixels” do mesmo para que possam servir como entrada à rede talâmica.

❖ Código

```
[P1,np]=coge_el_patron_s_i(random_presentation,np);
P=linear(0.4.*P1);
rib_graph=[];

iteraciones=input('Introduzca el numero de iteraciones ok ');

for iter=1:iteraciones
    close;
end;

show_graph=input('Escriba 1 para mostrar grafico normal y 0 para no mostrar ');
train_test=input('Escriba 0 para patrones normales y 1 para patrones parciales ');
graf_fin=input('Escriba 1 para mostrar grafico final y 0 para no mostrar ');

if (train_test==1) np=1;
end

for iter=1:iteraciones
    numero=0;
    it=it+1;
    if (mean(mean(activacion_reticular))>=0.5*limite_superior)
        preparado_patron=0;
    end

    if (mean(mean(activacion_reticular))<0.2*limite_superior)
        preparado_patron=preparado_patron+1;
    end

    if (preparado_patron==8);

    if (train_test==0)
```

```

[P1,np]=coge_el_patron_s_i(random_presentation,np);
P=linear(0.15.*P1);
else
[P,np]=coge_el_patron_test(random_presentation,np);
end
%TODAS LAS SIGUIENTES INICIALIZACIONES PUEDEN BORRARSE PARA UN
FUNCIONAMIENTO MAS LIBRE
refractoriness=ones(filas_ret,columnas_ret);
ret_in_order=zeros(filas_ret,columnas_ret);
matriz_inhibiciones=zeros(filas_ret,columnas_ret);
activacion_reticular=zeros(filas_ret, columnas_ret);
activacion_reticular_base=zeros(filas_ret, columnas_ret);
activacion_reticular_inputs=zeros(filas_ret, columnas_ret);
Ih=zeros(filas_ret,columnas_ret);

Ihk=0.2*limite_superior.*ones(filas_ret,columnas_ret);
In=zeros(filas_ret,columnas_ret);
Ink=zeros(filas_ret,columnas_ret);
activacion_ret_old=0.0.*activacion_reticular;
new_refract=zeros(filas_ret, columnas_ret);
prob_max_ret=zeros(filas_ret, columnas_ret);
B=zeros(filas_ret, columnas_ret);
C=zeros(filas_ret, columnas_ret);
end
if (rem(iter,ceil(iteraciones/200))==0)
it
end
if (it>1)
[prob_rele, matriz_inhibiciones]=prob_rele_f(ret_in_order,wrt,P,matriz_inhibiciones,0.7);
activacion_reticular_inputs=estim_retic_f(prob_rele,wrt,filas_ret,columnas_ret);%activacion_reticular es la
matriz de la iteracion anterior

[prob_max_ret, ret_in_order, activacion_ret_old, refractoriness,incremento, new_refract]=
prob_max_ret_f(activacion_reticular, activacion_reticular_inputs, activacion_ret_old, prob_max_ret,
ret_in_order,latencia_inhibicion, refractoriness,incremento_old, limite_superior, new_refract,
latencia_refract);
prob_max_ret;

end
activacion_reticular_base=activacion_reticular_base+Ih.*refractoriness+In; %Es importante aplicar el
periodo refractario no solo a los parametros de disparo sino a la propia entrada y al Ih
activacion_reticular=activacion_reticular_inputs.*refractoriness+activacion_reticular_base+(inc_potencial_g
anadora).*(prob_max_ret>zeros(filas_ret, columnas_ret));
B=activacion_reticular;
[valor_prob_max_ret,x_ganadora, y_ganadora]=max2D(prob_max_ret);
Ihk(x_ganadora, y_ganadora)=0.0;
Ink(x_ganadora, y_ganadora)=0.05*limite_superior;

for i=2:filas_ret-1
for j=2:columnas_ret-1

if (activacion_reticular(i,j)>0.98*limite_superior)
Ihk(i,j)=0.0;
Ink(i,j)=0.05*limite_superior;

```

```

end;
if (activacion_reticular(i,j)<=(0.02+limite_inferior))
    Ihk(i,j)=0.2*limite_superior;
    Ink(i,j)=0.0;
end;
    In(i,j)=0.*Ih(i,j)-Ink(i,j)*(1-exp(-activacion_reticular(i,j)/0.14));
    Ih(i,j)=0.*Ih(i,j)+Ihk(i,j)*(exp(-activacion_reticular(i,j)/0.2));

    In(i,j)=((1-4*peso_entre_dendroden)*In(i,j)+peso_entre_dendroden*(In(i+1,j)+In(i-1,j)+In(i,j-
1)+In(i,j+1)));

end
end

C=B;
C(1,1)=(B(1,1)+B(1,2)+B(2,1))/3;
C(1,columnas_ret)=(B(1,columnas_ret-1)+B(2,columnas_ret)+B(1,columnas_ret))/3;
C(filas_ret,1)=(B(filas_ret-1,1)+B(filas_ret,2)+B(filas_ret,1))/3;
C(filas_ret,columnas_ret)=(B(filas_ret,columnas_ret)+B(filas_ret-1,columnas_ret)+B(filas_ret,
columnas_ret-1))/3;

C(1,:)=B(2,:);
C(filas_ret,:)=B(filas_ret-1,:);
C(:,1)=B(:,2);
C(:,columnas_ret)=B(:,columnas_ret-1);
    activacion_reticular=C;

if(graf_fin==1)
    for i=2:filas_ret-1
        for j=2:columnas_ret-1
            numero=numero+1;
            rib_graph(numero, iter)=activacion_reticular(i,j);
        end
    end
end %end graf_fin
if (show_graph==1)
    %pause;
    graficoPCAsincosc_s_j(matriz_inhibiciones,activacion_reticular, end_flag, it, 0.5,
filas_rel,columnas_rel,P,P1,ret_in_order, prob_max_ret, prob_rele,refractoriness, np);
    %pause;
end

end %end for iter
if (graf_fin==1)
figure
ribbon(rib_graph);
view(110,50)
axis([0,(filas_ret-1)*(columnas_ret-1),0, iter,0,limite_superior]);
wtr_end=wtr;
wrt_end=wrt;
end
rib_graph=[];

```

- *Programa: GraficoPCAsincosc\_s\_i*

Este programa é igual ao programa GraficoPCAsinc\_osc, mas está adaptado a modelagem da estabilização de imagens na retina. Assim as matrizes P1 e P2 vão conseguir representar em verde o padrão e, em vermelho, a fração do padrão que realmente será processada.

❖ Código

```
function graficoPCAsincosc_s_i(matriz_inhibiciones,A, end_flag, iter,
factor_elevacion,filas_ret,columnas_ret,P,P1,ret_in_order, prob_max_ret, prob_rele,refractoriness, np)
D=A;
matriz_inh_graf= matriz_inhibiciones;
[filas_ret, columnas_ret]=size(A);
P2=(P1-P)>0.5.*ones(filas_ret, columnas_ret);
P2=P2-P;
elevacion=factor_elevacion.*ones(filas_ret-2, columnas_ret-2);
D(:,1)=[];
matriz_inh_graf(:,1)=[];
D(1,:)=[];
matriz_inh_graf(1,:)=[];
D(filas_ret-1,:)=[];
matriz_inh_graf(filas_ret-1,:)=[];
D(:,columnas_ret-1)=[];
matriz_inh_graf(:,columnas_ret-1)=[];

pause(0.0001);
subplot('position', [0.30,0.30,0.4, 0.4])
colormap(copper);
mesh(5.*D+elevacion);
hold on
surf(5.*D+elevacion), shading interp;
hold on;
media=mean(mean(matriz_inhibiciones));
contour(D)
title(strcat('Iteration number ', num2str(iter), ' Pattern number ', num2str(np)));

grid off;
hold off;
axis([1,filas_ret-2,1,columnas_ret-2,0,2.5]);
view(-37.5, 20);

subplot('position', [0.05,0.77,0.16, 0.16])
```

```

grafico_red(P2);
xlabel('Pattern');

subplot('position', [0.05,0.45,0.16, 0.16])%El 42 inicial fue cambiado a 47
grafico_red(logsig(15.*(matriz_inhibiciones-0.6.*ones(size(matriz_inhibiciones)))));
xlabel('Inhibitions');

subplot('position', [0.77,0.45,0.16, 0.16])
grafico_red(3000.*(~refractoriness));
xlabel('Refractoriness');

subplot('position', [0.05,0.10,0.16, 0.16])
grafico_red(prob_rele,1,0);
xlabel('Relay neurons');

subplot('position', [0.77,0.77,0.16, 0.16])
grafico_red(2000.*prob_max_ret);
xlabel('Winning neuron');

subplot('position', [0.77,0.1,0.16, 0.16])
grafico_red(ret_in_order);
xlabel('Reticular activ.');
```

## **APLICAÇÕES E PERSPECTIVAS FUTURAS:**

A estatística Euclidiana [24] que foi desenvolvida nesta tese mostrou-se uma ferramenta muito eficiente tanto para embasar de maneira simples as principais regras da lógica “Fuzzy” [35] como para entender a extração de componentes principais numa nova rede neural baseada no tálamo [32]. Sem esta nova ferramenta muitas das demonstrações (especialmente as referentes ao cálculo de componentes principais) teriam sido extremamente complicadas e obscuras.

Por outra parte, os desenvolvimentos realizados na área da lógica Fuzzy contribuirão para evitar os métodos de tentativa e erro no desenho “fuzzy” permitindo que, no futuro, os computadores realizem todas as fases deste desenho desde o início até o final a partir dos dados fornecidos.

A rede talâmica criada usando-se as ferramentas da estatística Euclidiana poderá, no futuro, ser implementada num circuito integrado analógico, onde o cálculo do neurônio de atividade máxima na camada competitiva seria realizado de forma parecida a como acontece na biologia. A soma ponderada das entradas que é o valor da atividade de cada neurônio seria multiplicada por uma voltagem oscilatória que atuaria simultaneamente em todos os neurônios. O primeiro neurônio que atingisse um determinado limiar seria o neurônio vencedor. Com isto poderiam-se conseguir velocidades extraordinárias de processamento num circuito integrado para compressão de dados.

As aplicações mais imediatas da rede talâmica poderiam estar na área de detecção de defeitos em peças mecânicas ou eletrônicas ou na manutenção

preventiva [34]. Tudo isto por causa da rede neural ser especialmente sensível a pequenos desvios da normalidade nos padrões que são apresentados para ela.

## ANEXO 1

### **Definição de espaço linear:**

Um conjunto  $U$  não vazio de objetos se chama espaço linear, se satisfaz os seguintes dez axiomas que são mostrados em três grupos.

#### *Axiomas de clausura ou fechamento*

AXIOMA 1. CLAUSURA OU FECHAMENTO DA ADIÇÃO: Para todo par de objetos  $x$  e  $y$  de  $U$  corresponde um objeto único de  $U$  que chamaremos de soma de  $x$  e  $y$  designado por  $x+y$

AXIOMA 2. CLAUSURA OU FECHAMENTO DA MULTIPLICAÇÃO POR NÚMEROS REAIS: Para todo  $x$  de  $U$  e todo numero real  $K$ , corresponde um objeto de  $U$  chamado  $Kx$ .

#### *Axiomas para a adição de números*

AXIOMA 3. LEI CONMUTATIVA. Para todo  $x$  e todo  $y$  de  $U$ , temos que  $x+y=y+x$ .

AXIOMA 4. LEI ASSOCIATIVA. Quaisquer que sejam  $x$ ,  $y$ ,  $z$  de  $U$  temos  $(x+y)+z=x+(y+z)$ .

AXIOMA 5. EXISTÊNCIA DE ELEMENTO ZERO. Existe um objeto em U, designado com o símbolo 0, tal que:  
 $x+0=x$  para todo x de U.

AXIOMA 6. EXISTÊNCIA DE OPOSTOS. Para todo x de U, o objeto  $(-1)x$  tem a propriedade:  
 $x+(-1)x=0$

Axiomas para a multiplicação por números

AXIOMA 7. LEI ASSOCIATIVA. Para todo A de U e todo par de números reais a e b, temos  
 $a(bx)=(ab)x$

AXIOMA 8. LEI DISTRIBUTIVA PARA A ADIÇÃO EM U. Para todo x e todo y de U e todo número real a, temos:  
 $a(x+y)=ax+ay$

AXIOMA 9. LEI DISTRIBUTIVA PARA A ADIÇÃO DE NUMEROS. Para todo x de U e todo par de números reais a e b, temos.  
 $(a+b)x=ax+bx$

AXIOMA 10. EXISTÊNCIA DE ELEMENTO IDÊNTICO. Para todo x de U, temos  $1x=x$

### **Definição de produto escalar ou interior:**

Um espaço linear real  $U$  tem um produto interior, se a cada par de objetos  $x$  e  $y$  corresponde um número real único  $(x,y)$  que cumpre com os seguintes axiomas, quaisquer que sejam  $x, y, z$  de  $U$  e para todos os valores reais  $c$ .

- 1)  $(x,y)=(y,x)$  (conmutatividade, ou simetria)
- 2)  $(x, y+z)=(x,y)+(x,z)$  (distributividade, ou linearidade)
- 3)  $c(x,y)=(cx,y)$  (associatividade, ou homogeneidade)
- 4)  $(x,x)>0$  se  $x \neq 0$  (positividade)

Um espaço linear com um produto interior se chama espaço real Euclidiano.

### ***Bibliografia.***

- [1] Artola, A., Brocher, S. & Singer W. (1990). Different voltage-dependent threshold for inducing long-term depression and long-term potentiation in slices of rat visual cortex. *Nature*, **347**:69-72.
  
- [2] Devijver, P.A., & Kittler J. (1987). *Pattern Recognition Theory and Applications*. NY: Springer Verlag.
  
- [3] Diamantaras K.I., Kung S.Y. (1996). *Principal Components Neural Networks*. John Wiley & Sons, Inc. pp.108.
  
- [4] Duda, R.O. & Hart P.E. (1973). *Pattern Classification and Scene Analysis*. John Wiley and Sons, Inc.
  
- [5] Finetti, Bruno de (1974). *Theory of Probability*. John Wiley & Sons, Ltd.
  
- [6] Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition (Second Edition)* San Diego: Academic Press.
  
- [7] Goldstein, M. (1981). Revising Previsions: A Geometric Interpretation. *Journal of the Royal Statistical Society*, **43**(2):105-130.
  
- [8] Haykin, S. (1999). *Neural Networks. A Comprehensive Foundation*.

Pentice Hall.

- [9] Hebb, D.O. (1949) *The organization of behavior*. New York: Wiley.
- [10] Hodgkin A.L., Huxley A. F. (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of physiology (London)* **111**:500-544.
- [11] Jaynes, E.T. (1979). Where do we stand on maximum entropy. in R. D. Levine and M.Tribus, Eds., *The Maximun Entropy Formalism*, MIT Press, Cambridge, Massachusetts.
- [12] Kohonen, T. Physiological interpretation of the self-organizing feature map. *Neural Networks*, **6**: 895-905.
- [13] Kohonen, T. The neural phonetic typewriter. *Computer*, **21**(3):11-22.
- [14] König, P., Engel,A.K. & Singer W. (1996). Integrator or coincidence detector? The role of the cortical neuron revised. *Trends in Neuroscience*, **19**(4): 130-137.
- [15] Kosko, B. (1983). *Neural Networks and Fuzzy Systems: A dynamical System Approach to Artificial Intelligence*. Prentice Hall, Engelwood Cliffs, NJ.

- [16] Llinas, R. & Jahnsen, H. (1982). Electrophysiology of mammalian thalamic neurons in vitro. *Nature*, **297**,:406-408.
- [17] Llinas, R.R. (1988). The intrinsic electrophysiological properties of mammalian neurons: Insights into Central Nervous System Function. *Science*, **242**:1654-64.
- [18] Llinas, R.R., Grace, A.A. & Yarom, Y. (1991). In vitro neurons in mammalian cortical layer 4 exhibit intrinsic oscillatory activity in the 10 to 50-Hz frequency range. *Proc. Natl. Acad. Sci.*, **88**:897-901.
- [19] Llinas, R., Ribary, U., Joliot, M. & Wang, X.J. (1994). Content and Context in Temporal Thalamocortical Binding. In G.Buzsaki et al. (Eds.). *Temporal Coding in the Brain* (pp. 151-72). Berlin: Springer-Verlag.
- [20] McClelland, J.L., Rumelhart, D.E. & The PDP Research Group. (1986). *Parallel distributed processing: Exploration in the microstructure of cognition*. Cambridge, MA: MIT Press.
- [21] McClelland, J.L., Rumelhart, D.E. (1988). *Explorations in parallel distributed processing*. Cambridge, MA: MIT Press.
- [22] Mulle C., Madariaga A.M., Deschenes M. (1986). Morphology and Electrophysiological Properties of Reticularis Thalami Neurons in Cat: In

Vivo Study of a Thalamic Pacemaker. *The Journal of Neurosciences*, 6(8):2134-2145.

- [23] Pinault, D. & Deschenes, M. (1992). Voltage dependent 40-Hz oscillations in rat reticular thalamic neurons in vitro. *Neuroscience*, 51:245-258.
- [24] Ropero Peláez, J. (1991). Una nueva formulación de la estadística desde el punto de vista de los espacios euclídeos multidimensionales y su aplicación a las redes neuronales. *Anales de Mecánica y Electricidad. (Revista de la Asociación de Ingenieros del ICAI)*, 68(4):43-49.
- [25] Ropero Pelaez, J. (1996a). A formal representation of thalamus and cortex computation. In: Eds. R. Moreno-Diaz; J. Mira-Mira. Brain processes, theories, and models: an international conference in honor of W.S. McCulloch 25 years after his death (pp. 154-163) Cambridge, Mass., MIT Press.
- [26] Ropero Pelaez, J. (1996b). A Model of information processing and learning through thalamus and cortex neural connectivity and their neuron specific properties. In: Eds. The Inst. of Electrical Engineers of Japan and Muroran Inst. of Technology, Proceedings of: International Workshop on Soft Computing in Industry 96 (pp. 139-147). Muroran, Japan.

- [27] Ropero Pelaez, J. (1996c). Several Hypotheses on Temporal Processing in Pyramidal Neuron's Dendrites. Proceedings of the First Online Workshop on Soft Computing. On the internet served by Nagoya University.
- [28] Ropero Pelaez, J. (1997). Plato's theory of ideas revisited. *Neural Networks. 1997 Special Issue*. **10**(7):1269-1288.
- [29] Ropero Peláez, J., Marcelo G. Simões (1999a). Pattern Completion Through Thalamo-Cortical Interaction. Proceedings of the International Jont Conference on Neural Networks 1999, Washington DC.
- [30] Ropero Peláez, J., Marcelo G. Simões (1999b). A Computational Model of Synaptic Metaplasticity. Proceedings of the International Jont Conference on Neural Networks 1999, Washington DC.
- [31] Ropero Peláez, J. (2000). Towards a neural network-based therapy for hallucinatory disorders.
- [32] Ropero Peláez, J., Marcelo G. Simões (2001a). The thalamus as a principal component neural network. Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, SCI 2001.10:353-358. Orlando (Florida).
- [33] Ropero Peláez, J., Newton Maruyama (2001b). The dynamics of

stabilized images in the retina contributes to support the hypothesis of principal components computation at the level of the thalamus. Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, SCI 2001.10:359-364. Orlando (Florida).

- [34] Ropero Peláez, J., Aguiar, M.A., Destro, R.C., Kovács, Z.L., Simões, M. G. (2001c). Predictive maintenance oriented neural network system. PREMON. Proceedings of the Industrial Electronics Society Conference. IECON 2001. Denver (Colorado)
- [35] Ropero Peláez, J., Marcelo G. Simões (2001d). New Geometrical Interpretations For Probability And Fuzy Logic. In Press.
- [36] Shannon,C.E. (1963).The mathematical theory of communication. Edited by C.E. Shannon and W. Weaver. University of Illinois Press, Urbana, IL, pp. 29-125.
- [37] Steriade, M. & Llinas, R.R. (1988). The Functional State of the Thalamus and the Associated Neuronal Interplay. *Physiol. Rev.*, **68**(3):649-739.
- [38] Steriade, M., Curró Dossy, R. & Contreras, F. (1993). Electrophysiological properties of intralaminar thalamocortical cells discharging rhythmic (40 Hz) spike-bursts at 1000 Hz during waking and rapid eye movement sleep. *Neuroscience*, **59**:21-31.

- [39] Taylor, J.G. & Alavi, F.N. (1993). Mathematical Analysis of a Competitive Network for Attention. In J.G.Taylor (Ed.). *Mathematical Approaches to Neural Networks* (pp.341-82). Amsterdam: Elsevier Science.
- [40] Terano, T., Asai, K. and Sugeno, M. (1992). *Fuzzy Systems Theory and its Applications*. Academic Press, San Diego, CA.
- [41] Thomson, A.M. & Deuchars, J. (1994). Temporal and spatial properties of local circuits in neocortex. *Trends in Neuroscience*, **17**(3):119-126.
- [42] Tou J.T. & González R.C. (1974). *Pattern Recognition Principles*. Reading ( MA) Addison-Wesley.
- [43] Wehr, M. & Gilles, L. (1996). Odour encoding by temporal sequences of firing in oscillating neural assemblies. *Nature*, **384**:162-165.
- [44] Zadeh, L.A. (1973). Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on Systems, Man, and Cybernetics*, **3**(1), 28-44.