IMPLEMENTAÇÃO DA LÓGICA FUZZY EM REDES NEURAIS ARTIFICIAIS E APLICAÇÕES EM BIOLOGIA

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida pelo Sr. Paulo Bernardo Blinder e aprovada pela Comissão Julgadora.

Campinas, 04 de Agosto de 1994

Orientador: Prof. Dr. Rodney Carlos Bassanezi

Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação (UNICAMP), como requisito parcial para a obtenção do título de Mestre em Matemática Aplicada na área de Análise Aplicada.

AGOSTO - 1994

.1 Agradecimentos

Embora em geral qualquer conjunto de pessoas seja contável, acredito que para fazer juz ao apoio recebido, seria necessário um número não enumerável de agradecimentos. Porém faço questão de re-lembrar *nominalmente* algumas delas, das quais terei prazer de recordar-me sempre que reler estas linhas.

.1.1 O Tao da Ciência

Todavia, nestes anos de pós-graduação , não deixaram de haver passagens antológicas. Algumas delas, conhecidas inclusive pelo meu Orientador, como a viagem para o Chile para apresentar trabalho no Congresso Internacional de Biomatemática, com apenas $US\$200$ no bolso (emprestados por ele !), como aquelas desconhecidas pelos resto da humanidade e que só dizem respeito a nossa intimidade como a que vou descrever: Numa tarde esperando carona para a USP em frente ao Museu de arte sacra do Coordenador do Grupo de Neuropsicolinguistica Cognitiva para uma reunião, na mão direita um *notebook* em cujo disquete se encontravam idéias para um artigo e o último trecho da tese, na mão direita um pedaço de pão que consistia o meu almoço, no bolso menos de $US\$5$, resultado de doação por parte de um amigo, no estomago e na cabeça uma dor resultante do sono, da fome, da angústia e da revolta, que só pôde ser contrabalançada pelo apoio recebido das pessoas que acreditavam em nosso trabalho, desde aficcionadas a Ciência como o rabino Michael Attias, até pessoas cientificamente ativas como o Coordenador do grupo de Ciência Cognitiva do Instituto de Estudos Avançados Henrique Schutzer del Nero, que além de sua atenção , dispensaram-me teto, comida, honra e o que mais tenha sido necessário nos momentos de maior necessidade. Ás pessoas supra citadas (que D"us as abençõe), o meu cordial e sincero muito obrigado.

.1.2 Ying

Faz-se necessário o agradecimento pessoal às seguintes pessoas:

Pelo seu apoio tanto técnico quanto pessoal, indispensável para a elaboração desta:

- *Prof.Dr. Rodney Bassanezi*, meu Orientador.
- *Prof.Dr. Alejandro B. Engel*, do Rochester Institute of Technology.
- *Prof.Dr. José Roberto Piqueira*, POLI-USP.
- *Prof.Dr. Osvaldo Frota Pessoa Jr.*, CLE-UNICAMP.
- *M.Sc. Eduardo Tavares Paes*, IO-USP.
- M. Lourdes e equipe, da Datilografia Especializada.
- Aos colegas do Grupo de *Ciência Cognitiva* do Instituto de Estudos Avançados da USP.
- Aos colegas do Grupo de *Ordem* e *Desordem* do Centro de Lógica e Epistemologia da UNICAMP.
- Aos amigos do Grupo de *Teoria da Ciência e Computação Cognitiva*, **FICÇÕES**, do referido Centro.

Pelo apoio pessoal, sempre necessário nas horas aflitas:

- À minha Familia.
- Henrique S. Del Nero e Familia.
- Écio de Jesus Nogueira e Família.

- Eduardo Passos Pereira.
- Atelmo A. Bald e Familia.
- Rabino Michael Attias e Familia.
- L.E. Marra.
- Marcos Jesus (vulgo "Argentino").
- Ricardo Kondo.

.1.3 Yang

E também, porque não dizer que a vida não é feita somente de flores ? Nossa homenagem também aos que, por um motivo ou por outro, estiveram conosco em alguma época e não se encontram mais ao nosso lado.

.1.4 Money, Money, Money...

Pela ajuda financeira:

- Ao **CNPQ**, pela bolsa de mestrado.
- A **FAEP**, pela bolsa de auxíliar de ensino.

Conteúdo

Capítulo 1

Teoria Clássica

1.1 Introdução Conceitual à Lógica

> *Lógica* é a ciência do raciocínio, demonstração , pensamento ou inferência.
>
> (Traduzido do Oxford English Dictionary)

Lógica pode ser intuitivamente definida como sendo o estudo dos métodos e princípios do raciocínio em todas as suas formas. Pode-se dizer também que a Lógica seria o estudo dos princípios formais de uma argumentação válida.

Portanto, a Lógica preocupa-se com o aspecto estrutural ou formal dos argumentos ao invés de preocupar-se com o conteúdo em si.

Vejamos, por exemplo, a seguinte frase:

Nova York é capital da Inglaterra ou Nova York é capital dos EUA.

Nova York não é capital da Inglaterra.

Logo, Nova York é capital dos EUA.

O argumento acima é considerado logicamente válido, pois a conclusão é deduzida à partir das premissas fornecidas, muito embora qualquer um com um mínimo de conhecimento em Geografia sabe que esta conclusão é incorreta.

Mais formalmente, diremos que a Lógica Clássica se ocupa do estudo das *proposições* .

Proposições são sentenças declarativas (em contraposição às sentenças interrogativas ou as imperativas). As sentenças acima fornecem-nos exemplos de proposições logicamente válidas. Alguns exemplos de sentenças que não são proposições válidas podem ser: "Onde fica Nova York ?" e "Estudarás Lógica".

A toda proposição lógica podemos atribuir o que chamaremos de *valor-verdade* da proposição , que deve ser ou *verdadeiro* ou *falso* (denotados por $\mathcal{V}$ ou 1, e $\mathcal{F}$ ou 0 , respectivamente).

1.2 Introdução Informal ao Cálculo Proposicional

Com algum "abuso de linguagem" podemos traçar um paralelo entre a "linguagem das expressões aritméticas" da algebra elementar e a "linguagem das proposições ", usada no *Cálculo Proposicional* . Na algebra elementar utilizamos as letras do alfabeto para denotar quantidades a valores numéricos, nas quais aplicamos operadores aritméticos ($+, -, \times, /$ etc.) para formar expressões a valores numéricos, cujo valor depende da construção da expressão. Por analogia, utilizamos no Cálculo Proposicional letras do alfabeto para representar proposições que são valorizadas como *verdadeiro* ou *falso* e aplicamos a elas operadores lógicos ($\neg, \wedge, \vee, \Rightarrow$ etc., que representam "não","e","ou","implicação ", respectivamente) para obter proposições mais complexas. Tais operadores são ditos *conectivos lógicos.*

Seguindo a analogia da linguagem, tomaremos como unidade básica as proposições simples ou elementares, que são aquelas que não podem ser decompostas em nenhuma outra proposição mais simples. A elas aplicaremos os conectivos lógicos, de modo a obter expressões cada vez mais complexas, ditas *sentenças compostas.*

Proposições básicas distintas serão representadas por letras maiúsculas distintas A,B, C

etc. Como no Cálculo Proposicional não estaremos preocupados com o conteúdo factual de cada proposição senão com o valor-verdade da mesma e com sua distinção de qualquer outra proposição básica, de modo que não será nosso objeto de estudo a estrutura interna precisa de cada proposição (que é a motivação básica do Cálculo de Predicados da Lógica Clássica). Apesar disto, as proposições lógicas elementares, pelo fato de serem expressões declarativas a respeito de determinado sujeito, geralmente possuem a seguinte forma

$$\text{“}x \text{ é } P\text{”} \tag{1.1}$$

Forma Geral das Proposições Lógicas Elementares

onde x é o sujeito e P é um predicado a respeito do mesmo. O valor-verdade de tal proposição será o julgamento da verdade ou falsidade de tal afirmação .

Podemos ter operadores lógicos de qualquer ordem (unários, binários, etc.) mas cada operador particular deverá ser aplicado a um número fixo de operandos. Por exemplo, a negação $\neg$ é um operador unário, $\wedge$ e $\vee$ são operadores binários etc. Convém notar, no entanto, que cada operando de um conectivo pode ser tanto uma proposição simples quanto uma proposição composta. Também por uma convenção de linguagem, usaremos letras itálicas maiúsculas para denotar proposições genéricas, as quais não estão fixas em um contexto ou ainda não estão associadas em particular a nenhuma proposição básica.

Dadas formulações distintas para a mesma proposição usaremos o símbolo $=$ para denotar sua equivalência. Por exemplo, se em determinado contexto as proposições A e $B \wedge C$ denotarem uma mesma proposição , denotaremos esta equivalência por $A = B \wedge C$. Note que $=$ não é um conectivo proposicional, senão uma notação.

Para nossa conveniência, formularemos mais rigorosamente nossa linguagem proposicional. Tal definição deve ter dois estágios; O primeiro é a *sintaxe* (ou gramática), que é o conjunto de regras que descreve a construção de todas as proposições possíveis. O segundo,

semântico (ou regras de interpretação), que é o conjunto de regras que atribuem significado a cada proposição definida por uma sintaxe.

No Cálculo Proposicional , a sintaxe implicita na descrição informal dada anteriormente é expressa pelas seguintes regras:

1. Os valores-verdade 0 (ou $\mathcal{F}$, denotando *falso*) e 1 (ou $\mathcal{V}$, denotando *verdadeiro*) são expressões válidas no Cálculo Proposicional .

2. As proposições básicas A, B, C etc. são expressões válidas no Cálculo Proposicional .

3. Dadas as proposições A, B, cada uma das seguintes proposições

$$\neg A, A \wedge B, A \vee B, A \Rightarrow B$$

é uma proposição composta, que representa uma expressão válida no Cálculo Proposicional .

4. Não há nenhuma outra proposição além das que possam ser determinadas pelas duas regras anteriores que sejam proposições válidas no Cálculo Proposicional .

Observa-se que com respeito a regra 3 anterior, que é possivel definirmos e usarmos outros conjuntos de conectivos lógicos tais como $\neg, \wedge, \vee$ ou $\Rightarrow$. É possível mostrar que os conjuntos de conectivos anteriores possuem a propriedade de que a composição dos conectivos contidos neles podem ser usados de modo a gerar todos os outros conectivos lógicos possíveis. Tais conjunto de conectivos lógicos que gozam da propriedade descrita acima são ditos *completos*.

Diremos que toda expressão sintaticamente válida segundo as regras anteriores, resumi-

damente falando, serão chamados de *fórmulas lógicas.*

No tocante à semântica, aproveitaremos aqui a abordagem informal para deixar sua definição precisa para a próxima seção.

Como havíamos dito anteriormente, estaremos apenas preocupados com o valor-verdade de uma determinada proposição . Neste sentido, cada conectivo representa uma operação lógica determinando o sentido de cada proposição composta em termo da relação entre seus operandos. Apresentaremos informalmente, a seguir, a interpretação semântica aproximada em linguagem natural dos cinco principais conectivos lógicos utilizados aqui:

$\neg$ *Negação ("não")*

Exemplo: Meu carro não é azul

$$\neg(\text{Meu carro é azul})$$

$\wedge$ *Conjunção ("e")*

Exemplo: Eu sou jovem e Eu sou ativo

$$(\text{Eu sou jovem}) \wedge (\text{Eu sou ativo})$$

$\vee$ *Disjunção ("ou")*

Exemplo: Ou Eu vou estudar ou Eu vou passear

$$(\text{Eu vou estudar}) \vee (\text{Eu vou passear})$$

$\Rightarrow$ *Implicação ("implica","se ... então ...")*

Exemplo: Se Eu ganhar na Loto então Eu ficarei rico

$$(\text{Eu ganhar na Loto}) \Rightarrow (\text{Eu ficarei rico})$$

$\Leftrightarrow$ *Bicondicional ("é equivalente à","se e somente se")*

Exemplo: Eu pagarei as contas se e somente se Eu ganhar dinheiro

$$(\text{Eu pagarei as contas}) \Leftrightarrow (\text{Eu ganhar dinheiro})$$

Assim como na álgebra elementar, podemos usar a precedência de parênteses e outras

regras para simplificar a notação escrita, e também omitir os parênteses em determinadas circunstâncias, como por exemplo, escrever $a + b/c + d * e$ ao invés de $((a + (b/c)) + d * e))$. Podemos aplicar regras similares e convenções de modo a simplificar a forma escrita das fórmulas lógicas, tais como:

1. Os parênteses externos de qualquer proposição podem ser omitidos.
2. A ordem de precedência dos conectivos é $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$ com $\neg$ com precedência mais alta. Qualquer par de parênteses implícito nesta ordem de precedência deve ser omitido. Portanto, podemos escrever $\neg$ A $\vee$ B ao invés de $(\neg A) \vee B$ mas *não* ao invés de $\neg(A \vee B)$ desde que $\neg$ possui uma precedência mais alta que $\vee$.
3. Por convenção , conectivos binários são associados sempre ao conectivo da esquerda. De modo que podemos escrever A $\vee$ B $\vee$ C ao invés de (A $\vee$ B) $\vee$ C.

Quando as regras acimas são aplicadas a qualquer proposição composta em particular, não obtemos uma nova proposição , senão uma representação mais simples da proposição original. Desta forma, a proposição

$$\neg A \vee B \vee \neg C \wedge D \Leftrightarrow (A \wedge C \Rightarrow \neg B)$$

será considerada logicamente identica à proposição :

$$((((\neg A) \vee B) \vee ((\neg C) \wedge D)) \Leftrightarrow ((A \wedge C) \Rightarrow (\neg B)))$$

Desta forma completamos nossa abordagem informal do Cálculo Proposicional . Ao longo deste trabalho e ainda neste capítulo veremos algumas aplicações do Cálculo Proposicional .

Existem duas abordagens possíveis para representar e resolver os problemas solúveis através do Cálculo Proposicional , que serão caracterizadas informalmente como"Abordagem do Fun-

cional Verdade" e"Abordagem Dedutiva". Na primeira abordagem valorizaremos as questões observando todos os valores-verdade que possam assumir todas as proposições básicas envolvidas, enquanto na segunda buscamos definir regras para que possamos derivar ou "deduzir" analiticamente os resultados em que estivermos interessados. Embora as duas abordagens sejam equivalentes, ou seja, tenham o mesmo potencial para a resolução de problemas, a segunda delas tem sido utilizada classicamente na Inteligência Artificial, e será nossa principal abordagem neste estudo.

Apresentamos a seguir a abordagem do funcional verdade e em seguida a abordagem dedutiva.

1.3 Significado Semântico dos principais conectivos

O sentido dos conectivos binários pode ser descrito através da tabela verdade. Tomemos por exemplo o conectivo $\wedge$. Dado que $\wedge$ é um conectivo binário, ele possui dois operandos, digamos A e B. Como cada um dos operandos pode assumir dois valores, então teremos que o lado esquerdo da tabela verdade possui quatro linhas. Observando então o lado direito da tabela verdade para $\wedge$, temos que ele vale $\mathcal{V}$ apenas para o caso em que tanto A quanto B possuem valor-verdade $\mathcal{V}$, ou seja, idêntico ao significado do "e" em linguagem natural.

É necessário ter em conta, que apesar de podermos construir todos os conectivos a partir de um conjunto de conectivos completo, é necessário observar qual é o conectivo correto a ser empregado. Um exemplo simples é a proposição composta " *Ou eu estou dormindo ou eu estou acordado*" pode ser que interpretada como $\vee$, mas os dois eventos acontecerem simultaneamente é impossível do ponto de vista formal , daí A e B possuirem o mesmo valor simultaneamente deve levar $F(A,B)$ em $\mathcal{F}$ na tabela verdade. Portanto, devemos ter que o

conectivo apropriado para a tradução semântica deste exemplo deve ser o "Ou-Exclusivo" ou "Não Equivalência".

Já o operador $\Rightarrow$ é usado em formas verbais como "Se A então B", onde A é chamado de *antecedente* e B é chamado de *consequente*. Se A é V então B determina a veracidade da expressão lógica, enquanto no caso em que A é falso a expressão é considerada correta. Expliquemos melhor a afirmação anterior. Consideremos a sentença

"Se as luzes da casa estão acesas então está havendo consumo de energia".

Portanto, se as luzes da casa estiverem acesas, a veracidade da afirmação será dada pelo valor-verdade do consumo de energia. No entanto, nada podemos afirmar a respeito da situação se as luzes não estiverem acesas, pois o fato de estar havendo consumo de energia não contribui para a veracidade ou falsidade da sentença. Por "*default*", assumiremos que a sentença assume o valor verdadeiro se o antecedente for falso.

A definição funcional de $\Rightarrow$, conhecido como *implicação material*, é perfeitamente válida, embora a correspondência entre ele e a estrutura "se ... então" em linguagem natural é mais fraca do que se assumíssemos a implicação como se fosse alguns dos outros conectivos lógicos binários. Falando especificamente, a conexão entre o antecedente e o consequente é muito mais fraca na interpretação direta do que o uso que fazemos dele na linguagem natural. Por Exemplo, consideremos a seguinte frase:

"Se F.Collor não é mais presidente então Itamar Franco é o presidente".

É natural esperarmos que se Itamar Franco não seja o presidente, então Collor seja o presidente, e que as demais alternativas (que são *Collor é o presidente* e *Itamar é o presidente* e que nenhum dos dois seja presidente) sejam falsas. Mas esta não é a tabela verdade de $\Rightarrow$ senão também a do "Ou-Exclusivo".

Convém notar que o conectivo $\Rightarrow$ tem boas propriedades lógicas assim como foi definido, mas é necessário acautelar-se destes possíveis significados semânticos intuitivos.

1.4 A Abordagem do Funcional de Veracidade

Já havíamos estipulado anteriormente que cada proposição possui um valor-verdade ($\mathcal{V}$ ou $\mathcal{F}$). Na abordagem do Funcional de Veracidade assumiremos que toda vez que um conectivo é aplicado a seus respectivos operandos para formar uma proposição composta, o valor verdade de cada proposição formada é completamente determinado pelo valor verdade de seus operandos e pela identidade do respectivo conectivo. Ou seja, estipulamos que cada conectivo será um *funcional-verdade.* Assim, o significado de qualquer conectivo pode ser definido através de uma tabela que mostre, para cada um dos valores possíveis de seus operandos, o valor-verdade correspondente ao valor da proposição composta formada pela aplicação do conectivo aos seus respectivos operandos. Tal tabela é chamada de *tabela verdade.* Observe a tabela verdade para o conectivo de negação $\neg$:

A	$\neg A$
$\mathcal{F}$	$\mathcal{V}$
$\mathcal{V}$	$\mathcal{F}$

Sendo um conectivo unário, ele possui apenas um operando A, podendo assumir apenas dois valores possíveis, $\mathcal{F}$ ou $\mathcal{V}$. Sua tabela verdade possui apenas duas linhas, uma para cada valor de A possível. Cada entrada no lado direito representa o valor de $\neg A$ correspondente ao valor de A à sua direita.

O significado de qualquer outro conectivo lógico pode ser descrito por meio de tabelas verdade.

1.4.1 Outros conectivos lógicos unários e binários

Antes de explicarmos semanticamente os resultados das tabelas verdade definidas para os conectivos clássicos, nos preocuparemos em resolver a questão de quantos conectivos binários e unários distintos podem ser definidos. Desde que cada conectivo é completamente descrito por sua tabela verdade, a questão é equivalente a reponder quantas tabelas verdade distintas podem ser construidas para um dado número de operandos.

Se considerarmos qualquer conectivo lógico unário como sendo uma função $F(A)$, cuja variável é o seu operando A, então o lado direito da tabela verdade define o valor da função para cada valor de seu operando A. A questão se reduz portanto à determinação de quantos modos diferentes pode-se preencher a coluna do lado direito. No caso de conectivos unários, desde que temos duas linha que podem assumir os valores $\mathcal{V}$ ou $\mathcal{F}$, é quatro ($A^2_{2,2}$). Portanto, além do operador unário de negação , existem outros três (F_1, F_2 e F_3), cujas tabelas verdade são

A	F_1	F_2	F_3	$\neg A$
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$
$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$

F_1 é o operador "identidade", que deixa seu operando invariante. F_2 e F_3 são operadores constantes, que associam os valores $\mathcal{V}$ e $\mathcal{F}$ a qualquer valor do operando. Apesar disto, o único operando de interesse é a negação .

Tal função F é usualmente chamada de *função lógica* e seus argumentos de *variáveis lógicas*. Dado que cada variável lógica pode assumir dois valores distintos, n variáveis lógicas podem assumir 2^n valores lógicos possíveis, que correspondem ao número de colunas da tabela verdade. Como para cada linha a função F pode ser definida de dois modos distintos, teremos então 2^{2^n} conectivos n-ários distintos.

1.4.2 Um exemplo: Conectivos binários

Conforme vimos acima, conectivos binários podem ser definidos através de uma função $F(A, B)$, onde A e B denotam variáveis lógicas que podem possuir dois valores-verdade cada uma. Logo, teremos 4 possibilidades possíveis de combinações entre os operandos. Assim sendo, a tabela verdade de um conectivo binário possui quatro linhas, de modo que teremos de escolher quatro valores-verdade no lado direito da tabela verdade para definir uma função lógica. Dado que cada valor-verdade pode ser tomado de duas formas possíveis, teremos $2^4 = 16$ diferentes tabelas verdade possíveis, e portanto, 16 conectivos lógicos distintos podem ser definidos.

Tabela 1.1 *Conectivos lógicos possíveis em duas variáveis*

A	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	Nome de Função	Símbolo	Outros nomes	Outros símbolos
B	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	adotado	adotado	utilizados	utilizados
	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	Função Nula	0	Falsum	$F, \bot$
	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	N-OR	NOR(A, B)	Função de Pierce	$A \downarrow B, A \veebar B$
	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	Inibição	$A \nleftarrow B$	Desig. Própria	$A > B$
	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	Negação	$\neg B$	Complemento	$\overline{B}, \sim B, B^0$
	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	Inibição	$A \not\Rightarrow B$	Desig. Própria	$A < B$
	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	Negação	$\neg A$	Complemento	$\overline{A}, \sim A, A^0$
	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	Ou-Exclusivo	$A \oslash B$	Não Equivalência	$A \not\equiv B, A \oplus B$
	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	N-AND	NAND(A, B)	Conectivo de Sheffer	$A \barwedge B$
$\wedge$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	Conjunção	$A \wedge B$	Função "E"	$A \& B, A \times B$
$\Leftrightarrow$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	Bicondicional	$A \Leftrightarrow B$	Equivalencia	$A \equiv B$
	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	Identidade	A	Afirmação	
	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	Implicação	$A \Leftarrow B$	Condicional	$A \subset B, A \geq B$
	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	Identidade	B	Afirmação	
$\Rightarrow$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	Implicação	$A \Rightarrow B$	Condicional	$A \supset B, A \leq B$
$\vee$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	Função "OU"	$A \vee B$	Disjunção	$A + B$
	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	Função 1	1	Verum	$\mathcal{V}$

1.5 Tabela Verdade para Expressões Complexas

Como havia sido indicado anteriormente, os conectivos proposicionais podem ser utilizados para construir proposições de complexidade arbitrária, considerando-se pelo número de conectivos e pelo número de proposições básicas que ela contêm, ainda que os operando(s)

de qualquer conectivo possam ser eles mesmos proposições compostas. Para uma proposição complexa é útil poder construir uma tabela verdade que defina para cada possível combinação dos valores dos operandos, da mesma forma que as tabelas verdade dos conectivos básicos básicos foram contruidas. Tais tabelas verdade podem ser construídas por passos, usando as tabelas verdade básicas. O procedimento de construção pode ser melhor ilustrado por meio de um exemplo. Suponha que desejamos construir uma tabela verdade para a proposição complexa P cuja definição é dada por

$$P = ((A \vee B) \wedge C) \Rightarrow ((\neg A) \vee (B \vee C)) \tag{1.2}$$

Observemos primeiramente que apenas três proposições básicas (A,B e C) foram utilizadas na construção de P e portanto a tabela verdade de P deve ter $2^3 = 8$ linhas, representando todas as combinações de valores possíveis de suas proposições básicas. Nosso objetivo é construir a partir destes valores uma coluna que exiba o valor da proposição P para cada uma das oito combinações possíveis. Como P é da forma $A_1 \Rightarrow A_2$, com $A_1 = (A \vee B) \wedge C$ e $A_2 = \neg A \vee (B \wedge C)$, podemos usar a tabela verdade definida para $\Rightarrow$ para obter os valores da coluna de P a partir das colunas de A_1 e A_2. De modo análogo, podemos utilizar a tabela verdade de $\wedge$ para construir a coluna referente a A_1 se construirmos as colunas para $A \vee B$ e C. Podemos continuar este processo de recorrência até que cada operando seja uma das três proposições básicas A, B e C. Iniciando com as três proposições básicas e trabalhando na "direção oposta", teremos uma lista das proposições que compõem P de modo que cada proposição seja A, B ou C ou seja formada pela aplicação de conectivos simples às proposições operandas que ocorram anteriormente na lista. Assim a tabela verdade construida para P é

A	B	C	(A∨B)	((A∨B)∧ C	¬ A	(B ∧ C)	((¬ A)∨ (B ∧ C))	((A∨B)∧ C ⇒ (¬ A)∨ (B ∧ C))
1	2	3	4	5	6	7	8	9
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$
$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$
$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$
$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$
$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$
$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$
$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$

Tabela 1.2. Tabela verdade para ((A∨B)∧ C ⇒ ((¬ A)∨ (B ∧ C)).

Como observamos acima, as colunas de 1 a 3 dão os valores-verdade possíveis para as proposições básicas, as colunas de 4 a 8 dão os valores-verdade para proposições complexas que compõem P, e a coluna 9 nos dá o valor-verdade de P.

O exemplo acima ilustra como podemos utilizar as tabelas verdade básicas para obter e avaliar *qualquer* proposição no Cálculo Proposicional .

1.6 Propriedades Lógicas Fundamentais das Proposições

Podemos usar a definição dos conectivos funcionais por meio de sua definição funcional associado às técnicas de tabela verdade de modo a definir e caracterizar algumas das principais propriedades lógicas das proposições.

Diremos que a proposição A é uma *tautologia* (logicamente válida) se A possuir valor verdade $\mathcal{V}$ para *qualquer* combinação de valores das proposições básicas envolvidas em sua construção . Analogamente, diremos que A é uma *contradição* (lógica) se sua avaliação verdade for $\mathcal{F}$ para *qualquer* combinação de valores das proposições básicas que a constituem. É aparente que, pela definição verdade de ¬, A é uma tautologia se, e somente se, $\neg A$ é uma contradição .

Diremos que A é lógicamente *consistente* se ela não for uma contradição , isto é, se existir pelo menos um valor $\mathcal{V}$ em sua tabela verdade para A.

Dadas as proposições A e B, se a proposição composta $A \Rightarrow B$ for uma tautologia, então diremos que *A implica logicamente B* ou que *B é consequência lógica de A*. De modo análogo, se $A \Leftrightarrow B$ for uma tautologia, diremos que *A e B são logicamente equivalentes.*

1.7 Dedução Lógica

Com os conectivos proposicionais anteriormente descritos e com as noções de tautologia e contradição é possível a inferência dedutiva baseada em sentenças em linguagem natural, usando tabelas verdade. Segue-se dois exemplos de aplicação destes conceitos: verdade das expressões compostas, dadas premissas e análise de consistência.

1.7.1 Verdade das Expressões Compostas, dadas premissas

Dado um conjunto de premissas assumidas como verdade, podemos determinar a verdade de expressões compostas em termos de expressões simples pela aplicação direta da análise da tabela verdade da expressão. O primeiro passo é a tradução da expressão para a forma simbólica e então utilizar a técnica da tabela verdade para determinar a veracidade da expressão. Consideremos o seguinte conjunto de premissas:

1. A campainha está tocando.
2. Não há ninguem na porta.
3. O botão está quebrado.

Avaliemos a veracidade da expressão: *A campainha está tocando se e somente se houver alguém na porta ou o botão estiver quebrado.* Representemos por A,B e C as expressões A campainha está tocando, Há alguém a porta e O botão está quebrado, respectivamente. A tabela verdade para a frase acima é

A	B	C	(B∨C)	A ⇔ (B∨ C)
$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$

de onde concluimos que a expressão é verdadeira.

1.7.2 Análise de Consistência

Podemos utilizar também a técnica de tabelas verdade para analisar um conjunto de expressões compostas em sua consistência. Um conjunto de expressões (proposições) $A_1, A_2, ..., A_n$ serão ditas *consistentes entre si* se, e somente se, a proposição formada pela conjunção delas $A_1 \wedge A_2 \wedge ... \wedge A_n$ for uma proposição consistente (tautológica).

Por exemplo, três sujeitos são acusados de um crime: a Amante, o Bêbado e o Caseiro. Seus depoimentos são os seguintes:

A Amante diz: "O Bêbado o matou; o Caseiro é inocente".

O Bêbado diz: "Se a Amante é culpada então o Caseiro também é".

O Caseiro diz: "Eu não o matei; Algum dos outros o matou".

Analisemos a consistência Lógica das afirmações acima. As afirmações acima podem ser escritas simbólicamente como

$$D_A = \neg B \wedge C$$
$$D_B = \neg A \Rightarrow \neg C$$
$$D_C = C \wedge (\neg B \vee \neg A)$$

onde A, B e C representam a Amante é inocente, o Bêbado é inocente e o Caseiro é inocente, respectivamente.

A tabela verdade para $D_A \wedge D_B \wedge D_C$ fica

A	B	C	D_A	D_B	D_C	$D_A \wedge D_B \wedge D_C$
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$
$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$
$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$
$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$
$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$
$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$
$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$

Como existe pelo menos uma linha com $\mathcal{V}$ na coluna final, teremos precisamente que apenas esta linha (6) é consistente, ou seja, existe pelo menos um conjunto de valorização das afirmações que torna todas as afirmações consistentes entre si. Quanto ao crime podemos então concluir daí, baseado nos valores de A, B e C, que a Amante e o Caseiro são inocentes e que o Bêbado é culpado.

Convém notar que a resolução deste tipo de problema pelo método da tabela verdade é razoável apenas para pequenos problemas, pois o tamanho da tabela verdade cresce exponencialmente com o número de afirmações envolvidas. Para problemas maiores, teremos que utilizar outras ferramentas que serão descritas neste trabalho.

1.8 Resultados Gerais do Sistema Funcional de Veracidade

Com base no que foi exposto anteriormente, relacionaremos agora os principais resultados gerais para a formulação Funcional de Veracidade no Cálculo Proposicional . Para tanto, apresentaremos mais algumas definições úteis.

Introduziremos a notação de tautologia, denotada por $\models$. Note que do mesmo modo que $=,\models$ *não* é um conectivo no Cálculo Proposicional senão uma notação convencionada.

Diremos que uma tabela verdade é uma *tabela verdade básica* se ela for constituída apenas de proposições básicas, ou seja, de proposições atômicas e por conectivos. A tabela 1.1 é um exemplo de tabela verdade básica, enquanto a tabela 1.2 não o é.

Diremos que duas proposições A e B *possuem a mesma tabela verdade* se tomando suas tabelas verdade básicas, suas colunas forem idênticas, ou seja, para cada linha suas entradas forem ambas $\mathcal{V}$ ou $\mathcal{F}$.

Para uma determinada afirmação da forma $A \Leftrightarrow B$, se a valoração da afirmação for $\mathcal{V}$, então concluímos (a partir da tabela verdade de $\Leftrightarrow$) que A e B possuem os mesmos valores verdade.

Diremos que as proposições A e B *são lógicamente equivalentes* se elas possuirem a mesma tabela verdade. Isto segue da observação do resultado da linha anterior e da definição de equivalência lógica.

A relação de equivalência lógica é *reflexiva, simétrica e transitiva*, sendo portanto uma relação de equivalência no sentido matemático clássico. Formalmente:

a. qualquer proposição A é logicamente equivalente a si mesma.

b. Se A é logicamente equivalente à B, então B é logicamente equivalente a A.

c. Se A é logicamente equivalente à B, e B é logicamente equivalente à C, então A é logicamente equivalente à C

1.8.1 Inferência na Lógica Clássica

As tautologias são utilizadas na Lógica Clássica para fazer inferências dedutivas. Neste sentido, também são chamadas de *regras de inferência.*

O exemplo de tautologia mais frequentemente utilizado para inferência é denominada *Modus Ponens*, que é uma regra de inferência cujo enunciado formal é: *Dadas duas proposições verdadeiras A e $A \Rightarrow B$ (premissas), então podemos concluir daí o valor da proposição B (conclusão).* Demonstraremos

formalmente este resultado a seguir, utilizando as técnicas e os resultados anteriores.

Teorema 1.1. *(Modus Ponens)*

Para as proposições arbitrárias A e B, se A é uma tautologia e A logicamente implica B, então B também será tautologia, isto é, se $\models A$ e $\models A \Rightarrow B$ então $\models B$.

Demonstração : Consideremos uma linha arbitrária da tabela verdade primitiva de $A \Rightarrow B$. Como $\models A$, A possui valor $\mathcal{V}$ nesta (e nas outras) linhas. Da tabela verdade da implicação , temos que o valor lógico das proposições B e $A \Rightarrow B$ é o mesmo para esta linha da tabela verdade . Mas por hipótese $\models A \Rightarrow B$, então tanto $A \Rightarrow B$ quanto B possuem valor $\mathcal{V}$ nesta linha. Estendendo o resultado para as demais linhas da tabela verdade , temos que toda entrada da coluna de B será $\mathcal{V}$, logo $\models B$. **c.q.d.**

1.8.2 Outros esquemas de inferência relevantes

Existem muitos outros esquemas de inferência possíveis do ponto de vista lógico. Porém, além de *Modus Ponens*, os mais utilizados na prática são o *Modus Tollens* e o *Princípio de Resolução de Robinson.* Suas consequências são enormes na Inteligência Artificial Tradicional uma vez que são a base do esquema de *encadeamento para trás* (que será discutido em (1.10.3)) e da programação lógica e da implementação da linguagem de quinta geração para Inteligência Artificial, o *PROLOG* (PROgramming LOGic, implementado originalmente por Kowalsky). O enunciado formal deles é o seguinte:

Modus Tollens $\models A \Rightarrow B, \models \neg B$, então $\models \neg A$.

Princípio de Resolução de Robinson $\models A \vee B, \models \neg A \vee C$, então $\models B \vee C$

onde A, B e C são proposições do Cálculo Proposicional .

1.9 A Abordagem do Funcional de Veracidade

Já havíamos estipulado anteriormente que cada proposição possui um valor-verdade ($\mathcal{V}$ ou $\mathcal{F}$). Na abordagem do Funcional de Veracidade assumiremos que toda vez que um conectivo é aplicado a seus respectivos operandos para formar uma proposição composta, o valor verdade de cada proposição formada é completamente determinado pelo valor verdade de seus operandos e pela identidade do respectivo conectivo. Ou seja, estipulamos que cada conectivo será um *funcional-verdade*. Assim, o significado de qualquer conectivo pode ser definido através de uma tabela que mostre, para cada um dos valores possíveis de seus operandos, o valor-verdade correspondente ao valor da proposição composta formada pela aplicação do conectivo aos seus respectivos operandos. Tal tabela é chamada de *tabela verdade*. Observe a tabela verdade para o conectivo de negação $\neg$:

A	$\neg A$
$\mathcal{F}$	$\mathcal{V}$
$\mathcal{V}$	$\mathcal{F}$

Sendo um conectivo unário, ele possui apenas um operando A, podendo assumir apenas dois valores possíveis, $\mathcal{F}$ ou $\mathcal{V}$. Sua tabela verdade possui apenas duas linhas, uma para cada valor de A possível. Cada entrada no lado direito representa o valor de $\neg A$ correspondente ao valor de A à sua direita.

O significado de qualquer outro conectivo lógico pode ser descrito por meio de tabelas verdade.

1.9.1 Outros conectivos lógicos unários e binários

Antes de explicarmos semanticamente os resultados das tabelas verdade definidas para os conectivos clássicos, nos preocuparemos em resolver a questão de quantos conectivos binários e unários distintos podem ser definidos. Desde que cada conectivo é completamente descrito por sua tabela verdade, a questão é equivalente a reponder quantas tabelas verdade distintas podem ser construídas para um dado número de operandos.

Se considerarmos qualquer conectivo lógico unário como sendo uma função $F(A)$, cuja variável é o seu operando A, então o lado direito da tabela verdade define o valor da função para cada valor de seu operando A. A questão se reduz portanto à determinação de quantos modos diferentes pode-se preencher a coluna do lado direito. No caso de conectivos unários, desde que temos duas linha que podem assumir os valores $\mathcal{V}$ ou $\mathcal{F}$, é quatro ($A^2_{2,2}$). Portanto, além do operador unário de negação , existem outros três (F_1, F_2 e F_3), cujas tabelas verdade são

A	F_1	F_2	F_3	$\neg A$
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$
$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$

F_1 é o operador "identidade", que deixa seu operando invariante. F_2 e F_3 são operadores constantes, que associam os valores $\mathcal{V}$ e $\mathcal{F}$ a qualquer valor do operando. Apesar disto, o único operando de interesse é a negação .

Tal função F é usualmente chamada de *função lógica* e seus argumentos de *variáveis lógicas*. Dado que cada variável lógica pode assumir dois valores distintos, n variáveis lógicas podem assumir 2^n valores lógicos possíveis, que correspondem ao número de colunas da tabela verdade. Como para cada

linha a função F pode ser definida de dois modos distintos, teremos então 2^{2^n} conectivos n-ários distintos.

1.9.2 Um exemplo: Conectivos binários

Conforme vimos acima, conectivos binários podem ser definidos através de uma função $F(A,B)$, onde A e B denotam variáveis lógicas que podem possuir dois valores-verdade cada uma. Logo, teremos 4 possibilidades possíveis de combinações entre os operandos. Assim sendo, a tabela verdade de um conectivo binário possui quatro linhas, de modo que teremos de escolher quatro valores-verdade no lado direito da tabela verdade para definir uma função lógica. Dado que cada valor-verdade pode ser tomado de duas formas possíveis, teremos $2^4 = 16$ diferentes tabelas verdade possíveis, e portanto, 16 conectivos lógicos distintos podem ser definidos.

Tabela 1.1 *Conectivos lógicos possíveis em duas variáveis*

A	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	Nome de Função	Simbolo	Outros nomes	Outros simbolos
B	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	adotado	adotado	utilizados	utilizados
	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	Função Nula	0	Falsum	$F, \bot$
	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	N-OR	NOR(A,B)	Função de Pierce	$A \downarrow B, A \vee B$
	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	Inibição	$A \not\Leftarrow B$	Desig. Própria	$A > B$
	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	Negação	$\neg B$	Complemento	$\overline{B}, \sim B, B^0$
	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	Inibição	$A \not\Rightarrow B$	Desig. Própria	$A < B$
	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	Negação	$\neg A$	Complemento	$\overline{A}, \sim A, A^0$
	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	Ou-Exclusivo	$A \oslash B$	Não Equivalência	$A \not\equiv B, A \oplus B$
	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	N-AND	NAND(A,B)	Conectivo de Sheffer	$A \barwedge B$
$\wedge$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	Conjunção	$A \wedge B$	Função "E"	$A\&B, A \times B$
$\Leftrightarrow$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{V}$	Bicondicional	$A \Leftrightarrow B$	Equivalencia	$A \equiv B$
	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{F}$	Identidade	A	Afirmação	
	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	$\mathcal{V}$	Implicação	$A \Leftarrow B$	Condicional	$A \subset B, A \geq B$
	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{F}$	Identidade	B	Afirmação	
$\Rightarrow$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{V}$	Implicação	$A \Rightarrow B$	Condicional	$A \supset B, A \leq B$
$\vee$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{F}$	Função "OU"	$A \vee B$	Disjunção	$A + B$
	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$	Função 1	1	Verum	$\mathcal{V}$

1.10 Introdução à Álgebra Booleana

A Álgebra Booleana é a base da análise e projeto de circuitos eletrônicos digitais. Estes circuitos compreendem componentes tais como dispositivos de acumulação capazes de armazenar representações de dígitos binários tipo 0 e 1, e elementos lógicos que podem efetuar operações elementares com estes valores. Como a Computação inclui uma boa dose de eletrônica digital, a Álgebra Booleana constitui-se numa das ferramentas básicas para o projetos de computadores.

Na Lógica Clássica uma expressão é ou verdadeira ou falsa, podendo ser matematicamente representada por 1 ou 0, respectivamente. Muitos sistemas físicos podem ser representados ou modelados por lógicas de dois estados. Por exemplo, num sistema elétrico uma luz está ACESA ou APAGADA e uma chave está ou ABERTA ou FECHADA. O estado em um computador é descrito por um conjunto de dígitos binários armazenados em sua memória.

A matemática da lógica de dois estados, atualmente chamada de Lógica Booleana, foi desenvolvida pelo matemático George Boole em torno de 1840. Foi então aplicada a circuitos por Claude Shannon, matemático americano, na década de 30. Ele utilizou a Álgebra Booleana de dois elementos pois estava interessado em redes de chaveadores (reles) que poderiam estar em dois estados, ABERTO ou FECHADO.

1.10.1 Definições

Uma *Álgebra Booleana* é um conjunto de dois elementos, denotados por 0 e 1, juntamente com duas operações + e ., chamadas adição e multiplicação

, para as quais valem as seguintes regras:

$$0+0=0$$
$$0+1=1+0=1$$
$$1+1=1$$

e

$$0.0=0$$
$$0.1=1.0=0$$
$$1.1=1$$

Note que estas regras são idênticas àquelas do Cálculo Proposicional com a substituição de $\wedge$ por . e de $\vee$ por +.

Uma *variável boolcana* é um símbolo tal qual A,B ou C, representando um dos valores 0 ou 1. Uma *expressão booleana* é uma expressão contendo variáveis booleanas, as constantes 0 e 1, e operadores booleanos + e . . Assim como na álgebra normal, podemos definir funções , por exemplo:

$$F(X,Y)=X.Y$$

Como cada variável booleana pode assumir dois valores, uma função booleana pode ser expressa em termos de uma tabela verdade onde todos os valores possíveis das variáveis da função são tabelados juntamente com o valor da função . A tabela verdade da função acima fica

X	Y	$X.Y$
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 1.3. Tabela verdade da função booleana $X.Y$.

Computar os valores de expressões mais complexas não é difícil, mas é necessário fazer uso de valores intermediários. Por exemplo, para computarmos

a função

$$F(X,Y,Z) = X.Y + Y.Z$$

usando o expediente das tabelas verdade, teremos

X	Y	Z	$X.Y$	$Y.Z$	$F(X,Y,Z)$
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	1	0	1
1	1	1	1	1	1

Tabela 1.4. Tabela verdade da função booleana $X.Y + Y.Z$.

Isto mostra claramente a analogia com o uso de tabelas verdade no Cálculo Proposicional . Duas funções booleanas serão ditas *iguais* se, para as mesmas entradas das variáveis boolanas, elas produzirem as mesmas saídas. Na implementação de circuitos lógicos é frequentemente necessária a implementação de circuitos que realizem determinada função com o mínimo custo. Para fazer isto, tomamos uma função que seja igual à desejada, mas que minimize o número de operadores requeridos. Por exemplo, a função lógica

$$G(X,Y,Z) = (X + Y).(X + Z)$$

é igual à função

$$F(X,Y,Z) = X + Y.Z$$

como pode ser observado através da seguinte tabela verdade

X	Y	Z	$X+Y$	$X+Z$	$(X+Y).(X+Z)$	$Y.Z$	$X+Y.Z$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Tabela 1.5. Tabela verdade das funções booleanas $(X+Y).(X+Z)$ e $X+Y.Z$.

É interessante observar que enquanto a expressão da função G têm três operadores, a função F têm apenas dois, fornecendo o mesmo resultado lógico.

1.10.2 Propriedades Algébricas Básicas

Na seção anterior, observamos a analogia entre os operadores booleanos + e . com seus equivalentes no Cálculo Proposicional $\wedge$ e $\vee$. A terceira operação booleana, a *complementação* , é equivalente ao conectivo $\neg$. Ela é representada na Álgebra Booleana por uma barra sobre a variável ou sobre a expressão, como utilizaremos neste trabalho (P.ex.: $\overline{A}$), ou por um apóstrofe (como A').

Portanto, as funções booleanas definidas na seção anterior podem ser implementadas por elemento lógicos do tipo $\wedge$,$\vee$ e $\neg$. Como o mencionado previamente, não é usual o caso em que uma dada função é implementada como foi definida, senão por uma função equivalente que utilize menos conectivos, ou ainda por conectivos diferentes. As propriedades mais importantes da Álgebra Booleana são exibidas a seguir (conforme seu equivalente no Cálculo Proposicional).

Distribuição	$X.(Y+Z) = X.Y + X.Z$
	$X + Y.Z = (X+Y).(X+Z)$
Associação	$X + Y + Z = (X+Y) + Z = X + (Y+Z)$
	$X.Y.Z = (X.Y).Z = X.(Y.Z)$
Complementação	$X.\overline{X} = 0$
	$X + \overline{X} = 1$
	$\overline{\overline{X}} = X$
Comutação	$X + Y = Y + X$
	$X.Y = Y.X$
Zero	$X.0 = 0$
	$X + 1 = 1$
Identidade	$X + 0 = X$
	$X.1 = X$
Idempotência	$X + X = X$
	$X.X = X$
Absorção	$X + X.Y = X$
	$X.(X+Y) = X$
	$X + \overline{X}.Y = X + Y$
Lei de De Morgan	$\overline{X + Y + Z \ldots} = \overline{X}.\overline{Y}.\overline{Z} \ldots$
	$\overline{X.Y.Z \ldots} = \overline{X} + \overline{Y} + \overline{Z} \ldots$

Algumas destas relações são obtidas trivialmente à partir das tabelas verdade dos operadores booleanos dadas anteriormente.æ

1.11 Reticulados

Reticulado é uma estrutura matemática que incorpora a noção de ordem. Observaremos primeiramente o conceito de *ordem parcial.*

Seja R relação binária definida como $R : L \rightarrow O \subset L \times L$. Portanto os elementos de R são pares ordenados (a, b) de elementos de L. R será dita uma *relação de ordem parcial* em L se ela satisfizer as seguintes condições :

1.$(a, a) \in R$ para todo $a \in L$. (*reflexividade*)

2. Se $(a,b) \in R$ e $(b,a) \in R$ então $a = b$. (*Anti-simetria*)

3. Se $(a,b) \in R$ e $(b,c) \in R$ então $(a,c) \in R$. (*transitividade*)

À primeira vista, dada sua formulação generalizada, a definição anterior parece algo "nebulosa", mas se por exemplo denotarmos por $a \leq b$ o par $(a,b) \in R$ da relação , ela se torna mais visível:

1. $a \leq a$.

2. Se $a \leq b$ e $b \leq a$ então $a = b$.

3. Se $a \leq b$ e $b \leq c$ então $a \leq c$.

Assim, a relação $\leq$, a qual é reflexiva, anti-simétrica e transitiva, comporta-se de modo semelhante à relação de ordem aritmética usual, e a noção de ordem parcial é uma generalização da relação de ordem. Mais ainda, se $\leq$ satisfizer

4. Para todo $a, b \in L$, ou $a \leq b$, ou $b \leq a$.

então ela será dita *relação de ordem total.*

Se $\leq$ é uma relação de ordem parcial em L, então o par $(L, \leq)$ será dito um *conjunto parcialmente ordenado*, ou *poset.* Embora tal propriedade seja uma conjunção entre L e a relação de ordem parcial, é comum dizer que L seja um poset e que a relação de ordem parcial envolvida seja dada no contexto.

Quando $\leq$ for uma relação de ordem total, L é dito *totalmente ordenado.* Um exemplo de conjunto totalmente ordenado é Z, o conjunto dos inteiros relativos, com sua relação de ordem sendo a aritmética usual. O conjunto de todos os subconjuntos do plano é um poset sobre inclusão ($\subset$); todavia, não é totalmente ordenado.

A seguir definiremos o conceito de reticulado, após realizar as definições básicas necessárias a tal.

Definição : *(Limite Superior)*

Seja R um subconjunto parciamente ordenado de L e u um elemento de L tal que $r \leq u$ para todo $r \in R$, então u será um *limite superior* de R.

Definição : *(Supremo)*

Sejam R,L e u definidos como acima. u será o *menor limite superior* ou *supremo* de R se para todo limite superior u' de R valer $u \leq u'$.

Definição : *(Limite inferior e Infimo)*

De modo análogo, l será o limite inferior de R se $l \leq r$, para todo $r \in R$. Será chamado de $Infimo$ se para todo limite inferior de R, r' valer $r \geq r'$.

Notação : (Infimo e Supremo)

Abreviam-se os nomes $Infimo$ e $Supremo$ por Inf e Sup, respectivamente.

Observação : Se L $(L, \leq)$ for um conjunto totalmente ordenado, então todo subconjunto de R possui um Supremo e um Infimo.

Definição : *(Reticulado)*

Um conjunto parcialmente ordenado sera um *reticulado* se todo subconjunto binário de R possuir um Inf e um Sup.

Notação : (Inf e Sup de pares ordenados)

Denotaremos por $x \vee y$ o Sup de $\{x, y\}$, e por $x \wedge y$ o Inf de $\{x, y\}$.

Propriedades e Tipos de Reticulados

Um reticulado possui as seguintes propriedades :

Para cada par $\{x, y\}$ de elementos do reticulado L, valem para o Inf e Sup:

1. Comutatividade
2. Associatividade
3. Idempotencia
4. Absorção

As quatro propriedades acima não são todas as propriedades possíveis de um reticulado, senão definem o "conceito de reticulado". Mais especificamente, se L for um conjunto munido das operações binárias $\wedge$ e $\vee$ satisfazendo as quatro propriedades acima, então L será um reticulado sob a relação de ordem $\leq$ definida por $a \leq b$ se e somente se $b = a \vee b$.

Nota Importante: Considere o conjunto LC de todas as sentenças bem formuladas no calculo proposicional. Se identificarmos sentenças logicamente equivalentes, se $\vee$ denotar *ou* e se $\wedge$ denotar e, então $(L, \vee, \wedge)$ será um reticulado. Mais ainda, se p e q são duas sentenças bem formuladas, entao $p \leq q$ significa $q = p \vee q$.

Definição : *(Reticulado Distribuído)*

Um reticulado será dito *distribuído* se além das quatro propriedades acima ele satisfizer a lei da distribuitividade

5. Distributividade

O Cálculo Proposicional é um exemplo de reticulado distributivo. Geralmente, todo conjunto totalmente ordenado é um reticulado distributivo.

Definição : *(Reticulado limitado) ou (Reticulado com unidades)*

Um reticulado limitado que possui dois elementos, ditos 0 e 1, que são seu menor e maior elemento, respectivamente, neste reticulado. Do ponto de vista

algébrico, um reticulado em geral é dado pela tripla $(L, \vee, \wedge)$, enquanto um reticulado limitado é dado pela quintupla

$$(L, \vee, \wedge, 0, 1)$$

onde valem as propriedades acima, acrescidas da seguinte:

6. $x \wedge 0 = 0$ e $x \vee 1 = 1$.

Definição : *(Reticulado Complementado)*

Um reticulado L é dito complementado se para todo elemento de L existir um elemento x', chamado *complemento de* x tal que

$$x \vee x' = 1 \text{ e } x \wedge x' = 0.$$

Definição : *(Reticulado Completo)*

Um reticulado é dito *completo* se todo subconjunto R de L possuir um supremo, ou equivalentemente, possuir um ínfimo.

1.11.1 Álgebras Booleanas

Um reticulado distributivo, limitado e complementado é chamado de *Álgebra Booleana*. Pode-se mostrar que numa Álgebra Booleana o complemento é sempre único. Dado que a Álgebra Booleana possui duas operações binárias ($\wedge$ e $\vee$) e uma operação unária (o complemento $'$), e duas unidades (0 e 1), ela é algébricamente tratada como sendo a sextupla

$$(L, \wedge, \vee, ', 0, 1)$$

Uma Álgebra Booleana de dois pontos $0, 1$, com $0 \leq 1$, é uma Álgebra Booleana das mais importantes, desde que ela é a base lógica dos computadores digitais atuais, que serve de base para o processamento de informações .

Uma *Álgebra de Heyting* é uma sextupla

$$(L, \wedge, \vee, \Rightarrow, \mathbf{0}, 1)$$

onde $(L, \wedge, \vee, \Rightarrow, 0, 1)$ é um reticulado limitado e distributivo, e $\Rightarrow$ é uma operação binária, chamada *implicação* , em L satisfazendo as seguintes condições :

H1. $x \Rightarrow x = 1$

H2. $(x \Rightarrow y) \wedge y = y$

H3. $x \wedge (x \Rightarrow y) = x \wedge y$

H4. $x \Rightarrow (y \wedge z) = (x \Rightarrow y) \wedge (x \Rightarrow z)$

H5. $(x \vee y) \Rightarrow z = (x \Rightarrow z) \wedge (y \Rightarrow z)$

Se

$$(L, \wedge, \vee, ', \mathbf{0}, 1)$$

for uma Álgebra Booleana, e se $a \Rightarrow b$ for definido como sendo $a' \vee b$, então

$$(L, \vee, \wedge, \Rightarrow, 0, 1)$$

é uma Álgebra de Heyting. Note que para transformar uma Álgebra Booleana em uma Álgebra de Heyting, definimos a operação $\Rightarrow$ de modo análogo a equivalência lógica entre a implicação material

Se a então b

e a sentença

(não a) ou b

no Cálculo Proposicional.

1.11.2 Sistemas Especialistas Tradicionais

Os sistemas Especialistas Tradicionais considerados nesta seção são os sistemas de *produção de inferências*, usualmente referidos na literatura de inteligência Artificial como *Sistemas de Produção* . Seu objetivo é a produção de inferências não trivias sobre proposições que representem nosso estado de conhecimento da realidade.

Os sistemas de produção produzem inferências baseados em:

1. Conhecimento adquirido e representado através de regras e armazenados em uma base de conhecimento (BC);
2. Um ou mais mecanismos de inferência lógica (Tais como *Modus Ponens*, *Modus Tollens* e o Princípio de Resolução de Robinson);
3. Conhecimento a respeito dos valores verdade das variáveis de estado representativas das realidade.

A aplicabilidade desta classe de sistemas dá-se em problemas aonde:

- Existam relativamente poucas variáveis de estado envolvidas e estas, além de bem conhecidas devem ser objetivamente mensuráveis;
- As relações nômicas entre as variáveis de estado (regras) devem ser nítidas e conhecidas com exatidão (conhecimento total da realidade).

Note-se que a escolha do mecanismo de inferência apropriado dependerá de qual seja a aplicação desejada.

1.11.3 Estratégias de dedução clássicas dos sistemas de produção

Entre as estratégias de dedução (inferência ou *produção*) clássicas em sistemas de produção , destaquemos aqui as duas mais usuais:

- Encadeamento para frente
- Encadeamento para trás

No caso do encadeamento para frente, percorre-se o banco de regras sequencialmente, instanciando-se o valor da variável consequente como verdadeiro se o antecedente da regra for uma expressão verdadeira naquele estágio de conhecimento do sistema até que nenhuma regra mais possa ser utilizada.

Isto equivale à aplicação da regra de inferência de *Modus Ponens*, pois se for conhecido, a partir da realidade (ou por dedução durante o processo de inferência), que a variável A possua valor verdade verdadeiro e que como as regras contidas na base de conhecimento são sempre verdadeiras, então podemos deduzir através de

$$\frac{A,\ A \Rightarrow B}{B}$$

De modo análogo, teremos a aplicação de *Modus Tollens* na estratégia de encadeamento para trás, percorrendo-se os consequentes das regras da base de conhecimento e instanciando-se os valores verdades das cláusulas antecedentes.

Deste modo, podemos considerar os sistemas de produção como uma aplicação direta do processo dedutivo da Lógica Clássica.

Várias simplificações podem (ou foram feitas) na dinâmica dedutiva do processo de inferência, através da representação das proposições em termos de suas expressões booleanas, redução de complexidade e aumento da eficiência por meio da otimização das sentenças booleanas resultantes (maiores detalhes sobre esta abordagem encontram-se em (Cara e Blinder, 1993)) ou ainda por meio de estratégias de busca em grafos e árvores (Rich,1988; Winston,1988).

Note-se ainda que, apesar destas estratégias representarem uma otimização do ferramental dedutivo, elas **não alteram em absolutamente nada** sua essência, ou seja, não são alterados os processos formais de dedução e representação do conhecimento, apenas obtêm-se com isto melhor performance.

1.11.4 Agruras dos Sistemas Especialistas Tradicionais

Embora os sistemas especialistas tradicionais tenham uma vasta gama de aplicações , eles possuem severas restrições ao seu uso na maioria das aplicações práticas, por exemplo:

- As variáveis de estado devem ser nítidas, decidíveis entre verdadeiro ou falso. Isto implica que devemos ter pelo menos uma variável para representar cada categoria de valores reais que a variável de estado possa assumir; além disto, devemos ter uma regra para cada estado que esta variável possa assumir. Isto nos leva a um aumento exponencial das necessidades de armazenamento e processamento para a resolução de problemas reais.
- O processamento nos sistemas especialistas tradicionais é efetuado de modo sequencial, onde apenas uma regra é considerada a cada instante. Isto implica na necessidade da utilização restrita às variáveis não dependentes, o que nem sempre é possível em casos reais.
- Impossibilidade de representação de conhecimento adquirido de forma natural. Isto se dá basicamente por duas razões: As regras são inflexíveis linguísticamente, o que implica na necessidade de uma regra para cada valor linguístico possível de ser assumido pela variável (uma restrição praticamente impossível de ser cumprida) e do eventual funcionamento "ilógico" dos raciocínios naturais (que levam alguns sistemas formais a deduzirem "logicamente" através de suas regras de inferências lógicas, raciocínios absolutamente não intuitivos e anti naturais, causando assim discrepâncias entre o resultado esperado da inferência)...

Podemos ainda continuar nossa relação de restrições como o problema da resolução de conflitos entre duas regras de conhecimento conflitantes entre si, a não monotonicidade da Lógica Natural, a dificuldade de tratamento de contradições e tantos outros fenômenos correntes em situações usuais na prática. Note que estas restrições **não inviabilizam** a utilização de sistemas especialistas, senão

que *restringem* a classe de suas potenciais aplicações .

Mais ainda, existem estratégias para a manipulação de incerteza em Sistemas Especialistas (a segunda geração), em sua maioria utilizando os chamados *fatores de certeza* e inferência bayesiana para derivar o grau de "*probabilidade de verdade*" de determinada proposição .

Todavia, supondo que não precisemos discutir a questão clássica da validade e da aplicabilidade da inferência baysiana, a interpretação de tais resultados torna-se ambígua na maioria dos casos, como por exemplo a ilustração de Zadeh (em "Fuzzy Logic in the management of uncertainty" (Zadeh, 1993)):

João possui úlcera duodenal,(FC=0.3)

onde a interpretação do resultado (FC=0.3) poderia significar: a) Que João possui úlcera duodenal com grau 0.3 (?); b) Que a probabilidade (??, pois teríamos que supor que o experimento é *repetível*, e ainda por cima infinitas vezes) de que João possua úlcera duodenal é 0.3; c) Que a *possibilidade* que João tenha úlcera duodenal é 0.3.

Estenderemos aqui o exemplo de Zadeh, de modo a torná-lo mais didático ainda, realizando a prescrição de remédio para úlcera duodenal de João, dependendo de qual interpretação acima seja aceita. Se aceitarmos a interpretação (a) (derivado do Sistema especialista de Diagnóstico médico MYCIN), perscreveremos remédio em *grau 0.3* (tomar 3/10 da dose usual?); se aceitarmos (b), diremos a João para tomar a dose normal do remédio 3 a cada 10 dias (!!); Se aceitarmos a não aditividade e independência dos eventos da vida (e da clínica médica) então observaremos que a possibilidade (c) de que João possua úlcera duodenal é considerável, porém baixa, e buscaremos o valor da possibi-

lidade resultante da aplicação de outras regras, que em sendo *mais possíveis*, serão o diagnóstico *mais provável* para o caso, receitando assim o tratamento usual para as enfermidade de João (e, talvez, para a felicidade[1] de alguma Maria...).æ

[1]Felicidade é uma variável fuzzy (!!) (veja [Lógica Fuzzy como fator de síntese].

Capítulo 2

Representação do Conhecimento através da Teoria Fuzzy

Neste capítulo apresentaremos um resumo dos principais resultados e aplicações da Teoria Fuzzy aos sistemas especialistas no período de 1965 a 1993.

Não será nosso objetivo considerar todos os resultados, senão apenas os principais deles relacionados com os processos de inferência fuzzy, cujos enunciados necessários a este estudo serão o assunto deste capítulo.

Como exemplo de aplicação dos conceitos envolvidos neste capítulo, faremos uma representação dos conceitos de fitosociologia através da Teoria Fuzzy, a qual será utilizada como base para os exemplos de inferência nos capítulos seguintes.

2.1 Teoria de Conjuntos Fuzzy

A Teoria de Conjuntos Fuzzy é um dos elementos centrais a toda a Teoria Fuzzy. A elegância de suas operações , associada a uma grande simplicidade

lhe proporcionam o papel de servir como vínculo para diversos pontos distintos desta Teoria, entre os quais a Lógica Fuzzy.

Apresentaremos nesta seção um resumo dos principais elementos da Teoria de Conjuntos Fuzzy[1], necessários aos desenvolvimentos seguintes.

2.1.1 Conjuntos Fuzzy

Um conjunto fuzzy A de um conjunto universo U é definido como o conjunto de pares ordenados $\{u, \mu_A(u)\}$, $u \in U$, no qual $\mu_A(u)$ é o grau de pertinência do elemento u ao conjunto A e seu valor $\mu_A(u)$ reside no intervalo $[0,1]$. A função $\mu_A : U \rightarrow [0,1]$ é chamada *função de pertinência* do conjunto fuzzy A, e caracteriza o conjunto fuzzy A através da notação :

$$A = \int_U \mu_A(u)|u \tag{2.1}$$

ou, se U for finito, $U = \{u_1, u_2, \ldots, u_n\}$,

$$A = \sum_{i=1}^{n} \mu_A(u_i)|u_i = \sum_{i=1}^{n} \mu_i|u_i \tag{2.2}$$

O *suporte* $Supp\ A$ de um conjunto fuzzy A é o conjunto de pontos em U que possuem grau de pertinência não nulo.

A *altura* de um conjunto fuzzy A é o valor $\sup_U \mu_A(u)$. Um conjunto fuzzy é dito *normal* se sua altura for igual a 1.

[1] A despeito da observação feita por Kauffman, de que conjuntos são sempre ordinários e de que apenas podem existir subconjuntos fuzzy, será utilizado o termo *conjunto fuzzy* de modo indistinto como encontramos atualmente na literatura, em contraposição aos "subconjuntos fuzzy" da metade da década de 70.

Denotaremos o conjunto de todos os conjuntos fuzzy no conjunto universo U por $\mathcal{F}(U)$ (ainda encontra-se na literatura $\tilde{P}(U)$ ou ainda $2^{\mathcal{P}(U)}$).

2.1.2 Operações entre Conjuntos Fuzzy

Sejam A e B dois subconjuntos fuzzy do conjunto universo U, e assumindo que eles sejam caracterizados pelas funções μ_A e μ_B. Apresentaremos primeiramente as definições ligadas às relações de igualdade e inclusão:

Igualdade e Inclusão

Igualdade $(A = B)$ ocorre $\Leftrightarrow (\forall u \in U)[\mu_A(u) = \mu_B(u)]$.

Inclusão $(A \subseteq B)$ ocorre $\Leftrightarrow (\forall u \in U)[\mu_A(u) \leq \mu_B(u)]$.

Operações básicas usuais

Algumas das mais comuns operações usuais entre conjuntos fuzzy são definidas como segue:

- O *Complemento* de um conjunto fuzzy A, denotado por $\neg A, \overline{A}$ ou A', é definido pela função de pertinência

$$\mu_{\neg A}(u) = 1 - \mu_A(u) \tag{2.3}$$

- A *União* $A \cup B$ de dois conjuntos fuzzy A e B é definida como sendo o menor conjunto fuzzy que contenha A e B simultaneamente:

$$\mu_{A \cup B}(u) = \max(\mu_A(u), \mu_B(u)) = \mu_A \vee \mu_B \tag{2.4}$$

- A *Intersecção* $A \cap B$ de dois conjuntos fuzzy A e B é definida como sendo o maior conjunto fuzzy que contenha A e B simultaneamente:

$$\mu_{A \cap B}(u) = \min(\mu_A(u), \mu_B(u)) = \mu_A \wedge \mu_B \tag{2.5}$$

- O *Produto* $A.B$ de dois conjuntos fuzzy A e B é definido pela função de pertinência

$$\mu_{A.B}(u) = \mu_A(u).\mu_B(u) \tag{2.6}$$

Outras operações entre conjuntos fuzzy utilizadas

As seguintes operações entre conjuntos fuzzy não possuem operações equivalentes na teoria de conjuntos nítidos. Uma discussão mais detalhada sobre a semântica de tais operações será efetuada mais adiante, na seção referente à Lógica Fuzzy propriamente dita.

- *Soma-limitada* ($\oplus$) de dois conjuntos fuzzy A e B é definida pela função de pertinência

$$\mu_{A \oplus B}(x, y) = \min(1, \mu_A(x) + \mu_B(y)) \tag{2.7}$$

- *Potência de um conjunto fuzzy.* Seja A um conjunto fuzzy $A \subseteq U$. A m–ésima potência do conjunto fuzzy A^m é definida pela função de pertinência

$$\mu_{A^m}(u) = [\mu_a(u)]^m \ , \ \forall u \in U, \ m \in I\!R^+ \tag{2.8}$$

– *Normalização* , $norm(A)$, de um conjunto fuzzy A é definida pela função de pertinência

$$\mu_{norm(A)}(u) = \mu_A(u) / \max_{v \in A}(\mu_A(v)). \tag{2.9}$$

Este operador é utilizado para tornar conjuntos fuzzy não-normais em conjuntos fuzzy normais, assegurando que exista pelo menos um elemento u tal que $\mu_A(u) = 1$.

– *Dilatação* , $dil(A)$, de um conjunto fuzzy A é definida pela função de pertinência

$$\mu_{dil(A)}(u) = \sqrt{\mu_A(u)}. \tag{2.10}$$

O uso deste operador resulta num aumento do grau de pertinência dos elementos do domínio. Isto significa que quanto menor o grau de pertinência de um objeto x ao conjunto A, maior será sua pertinência no conjunto modificado, significando uma relaxação das restrições com relação ao conceito fuzzy A.

– *Contração* , $con(A)$, de um conjunto fuzzy A é definida pela função de pertinência

$$\mu_{con(A)}(u) = \mu_A^2(u). \tag{2.11}$$

Este operador funciona de modo inverso ao operador de dilatação . Sua utilização resulta num aumento da restrição dos graus de pertinências dos elementos do conjunto. A perda de grau de pertinência dos elementos será tanto menor quanto mais próximo de 1 for a pertinência do elemento.

– *Intensificação* , $int(A)$, de um conjunto fuzzy A é definida pela função de pertinência

$$\mu_{int(A)}(u) = \begin{cases} 2\mu_A^2(u) & \text{se } 0 < \mu(u) < 0.5 \\ 1 - 2(1 - \mu_A(u))^2 & \text{se } 0.5 < \mu(u) < 1 \end{cases} \quad (2.12)$$

O uso deste operador resulta numa diminuição do grau de pertinência dos elementos que forem menores que 0,5 e num aumento do grau de pertinência daqueles que forem maiores.

2.1.3 Operações Generalizadas com Conjuntos Fuzzy

Quando Zadeh elaborou a Lógica Fuzzy, seus principais operadores eram o máximo e o mínimo, respectivamente relacionados com os operadores *"OU"* e *"E"* da Lógica Clássica. Desta maneira assegurava-se que a Lógica Fuzzy seria uma extensão da Lógica Clássica. Porém os pesquisadores em Lógica Fuzzy observaram que o Raciocínio Aproximado poderia utilizar-se de uma série de outras semânticas relevantes para a conjunção e disjunção (Zimmerman e Zysno, 1980; Greco et ali., 1984; Greco et ali., 1987).

Como observamos portanto, as operações de intersecção , união e complementação podem ser definidas de mais de um modo, além daqueles definidos em 2.1.2 [2,111,112,181,189]. Os diferentes métodos são resultado de propriedades semanticamente desejáveis para os conjuntos fuzzy resultantes de tais operações .

Em [189], Yager definiu operadores generalizados de intersecção e união como sendo uma família de funções dependentes de parâmetro:

$$\mu_{A\cup_p B} = 1 - \min(1,((1-\mu_A(x))^p + (1-\mu_B(x))^p)^{1/p}), \quad p \geq 1 \quad (2.13)$$

$$\mu_{A\cap_p B} = 1 - \min(1,((\mu_A(x))^p + (\mu_B(x))^p)^{1/p}), \quad p \geq 1 \quad (2.14)$$

Como notado em [189], o tamanho do parâmetro p serve como medida de "quão rígidas" se requer que sejam as operações acima.

Normas T e Conormas T

Dubois e Prade [idem, 1982] introduziram o conceito de *Norma T* (norma triangular) e *Conorma T* (co-norma triangular) na literatura fuzzy, tomado da estatística [e Skliar, 1963], para lidar de um ponto de vista abstrato com estas outras semânticas das operações de conjunção e disjunção , constituindo-se na abordagem mais geral possível para definir-se as referidas operações .

As normas triangulares são definidas axiomaticamente como segue [143,144].

Normas T

Definição *Norma T*

Uma *Norma T* triangular é uma função real de dois argumentos cujo domínio é o reticulado unitário $[0,1]\times[0,1]$, e que satisfaz as condições :

a) $T(0,0) = 0$, $T(x,1) = T(1,x) = x$ (condições de contorno)
b) $T(x,y) \leq T(w,z)$ se $x < w$ e $y < z$ (monotonicidade)
c) $T(x,y) = T(y,x)$ (comutatividade)
d) $T(x,T(y,z)) = T(T(x,y),z)$ (associatividade)

Sejam A e B dois conjuntos fuzzy de um mesmo universo de interesse I. A função de pertinência da intersecção dos dois conjuntos ($\mu_{A\cap B}$) pode ser definida através de uma função N, definida por

$$\mu_{A\cap B} = N(\mu_A, \mu_B) \tag{2.15}$$

onde N é uma Norma T. É fácil ver que uma Norma T satisfaz as condições desejáveis á operação de intersecção de conjuntos fuzzy, ou ainda o "E" lógico. Portanto, a operação de intersecção de conjuntos fuzzy pode ser generalizada através da Norma T. Em particular, $\cap_p$ pode ser definida em termos da norma T correspondente.

Exemplo das principais Normas T

operador	norma
mínimo	$\min(x, y)$
produto	$(x.y)$
T_M	$\max(0, x + y - 1)$

A maior norma T é a norma do maxímo e a menor é

$$T_w(a, b) = \begin{cases} a, & \text{se } b = 1, \\ b, & \text{se } a = 1, \\ 0 & \text{caso contrário.} \end{cases} \tag{2.16}$$

Desigualdades importantes das Normas T - Para as Normas T valem as seguintes desigualdades:

$$T_w(x, y) \leq \max(0, x + y - 1) \leq x.y \leq \min(x, y) \tag{2.17}$$

Mais ainda, para qualquer Norma T vale

$$T_w(x, y) \leq T(x, y) \leq \min(x, y) \tag{2.18}$$

Normas T Arquimedianas - Uma Norma T arquimediana satisfaz também a seguinte propriedade

$$T(x,x) \leq x \tag{2.19}$$

seguindo daí que T deve ser da forma

$$T(x,y) = f^{-1}(f(x) + f(y)) \tag{2.20}$$

onde $f : [0,1] \rightarrow [0,\infty]$ e f^{-1} é a pseudo inversa de f. O produto é uma Norma T arquimediana, e pode ser calculado se f for a função logarítmo e f^{-1} for uma função potência. A função f é chamada de *gerador aditivo de* T.

Conorma T

Definiremos uma Conorma T da seguinte forma:

Definição Uma *Conorma* T triangular (ou *Norma* S) é uma função real S de dois argumentos cujo domínio é o quadrado unitário $[0,1]\times[0,1]$, e que satisfaça as condições :

a) $S(1,1) = 1$, $S(x,0) = S(0,x) = x$ (condições de contorno)
b) $S(x,y) \leq S(w,z)$ se $x < w$ e $y < z$ (monotonicidade)
c) $S(x,y) = S(y,x)$ (comutatividade)
d) $S(x,S(y,z)) = S(S(x,y),z)$ (associatividade)

É igualmente fácil ver que a Conorma T satisfaz as propriedades desejadas para a união de conjuntos fuzzy e para o "OU" lógico, generalizando-os portanto.

Em particular, $\cup_p$ pode ser definida em termos da norma S correspondente.

Correspondência entre as Normas T e S

Qualquer Norma S pode ser gerada a partir de uma Norma T utilizando-se a seguinte transformação

$$S(x,y) = 1 - T(1-x, 1-y) \tag{2.21}$$

Deste modo, a transformação entre as Normas T e suas Normas S associadas ficam da seguinte forma:

$$\begin{array}{rcll} \min(x,y) & \rightarrow & \max(x,y) & \\ x.y & \rightarrow & x+y-x.y & \text{(soma probabilística)} \\ \max(0,x+y-1) & \rightarrow & \min(1,x+y) & \text{(soma limitada)} \end{array}$$

2.1.4 Números Fuzzy

Números Fuzzy são conjuntos fuzzy definidos de forma especial no conjunto universo $\mathbb{R}$. A principal característica distintiva dos números fuzzy é a de que sua função de pertinência possui uma forma característica, correspondendo de alguma forma a propriedades semânticas desejáveis destas funções , e que podem ser representados de forma única através do conjunto reduzido de parâmetros utilizados para sua geração. A maioria dos números fuzzy pode ser definida através de uma quádrupla de números reais (a,b,c,d), onde a função de pertinência μ_N de um número fuzzy N deve ser contínua por partes e satisfazer as seguintes propriedades:

1. $\mu_N : \mathbb{R} \rightarrow [0,1]$ é contínua em $\mathbb{R}$;
2. $(\forall x \in (-\infty, c])[\mu_N(x) = 0]$;
3. μ_N é estritamente crescente no segmento $[c,a]$;

4. $(\forall x \in [a,b])[\mu_N(x) = 1]$;

5. μ_N é estritamente decrescente no segmento $[b,d]$;

6. $(\forall x \in [d,+\infty))[\mu_N(x) = 0]$.

A quantidade $(a+b)/2$ é chamada de *valor médio* do número fuzzy N. Quando $a = b$ então o número fuzzy N é caracterizado pela propriedade de ser *"aproximadamente a"*.

Os números fuzzy provêem um ferramental básico para a construção de modelos nas aplicações reais. Devido a isto, tem sido crescente o interesse e o número de publicações sobre o assunto.

Veremos adiante que os tipos de números fuzzy utilizados nas construções de modelos é que permitirão a garantia de satisfação de determinadas propriedades semânticas em modelos de inferência fuzzy usual.

2.2 Teoria da Possibildade

Uma função $\pi_X(u) : U \rightarrow [0,1]$ que associa a cada $u \in U$ um número, Possibilidade$(x = u)$, é chamado de função de *distribuição de possibilidade* para os valores da variável x. Sobre o conjunto de todos os subconjuntos de U, esta função define uma *medida de possibilidade* Π_X:

$$(\forall F \subseteq U)[\Pi_X(F) = \sup_{u \in F} \pi_x(u)] \tag{2.22}$$

Sob a condição $(\exists u \in U)[\pi_x(u) = 1]$, a medida de possibilidade Π_X satisfaz os seguintes propriedades:

1. $\Pi_x(\emptyset) = 0$, $\Pi_x(U) = 1$,

2. $\Pi_x(\cup_{i \in I} A_i) = \sup_{i \in I} \Pi_x(A_i)$ para algum conjunto de índices I.

Porém, podemos citar entre suas principais distinções das referidas medidas as seguintes propriedades:

- A medida de possibilidade **não** é aditiva. Isto significa que sua soma não está restrita a nenhum valor particular. Isto também não quer dizer que obteremos uma medida de probabilidade normalizada através da distribuição de possibilidade. Veremos algumas distinções adiante.

Procuramos considerar a Teoria da Possibilidade como unidade autônoma que possui seus próprios princípios. Estes princípios foram distribuídos e espalhados em diversos trabalhos, mas podemos considerar dois deles como básicos: o de Mínima Especificidade e o de Combinação / Projeção. Devemos notar que as medidas de possibilidade são um caso particular das medidas fuzzy de Sugeno (Banon, 1977) , entre as quais podemos distinguir as medidas de probabilidade, de credibilidade e um número de outras (para maiores detalhes sobre este assunto, existe extensa e acessível bibliografia tal como (Sugeno, 1974; Banon, 1977; Dubois e Prade, 1980; Bassanezi, 1987; Gerônimo, 1988; Barros, 1992).

Façamos, todavia algumas considerações a respeito das propriedades mais importantes desta medida, e de suas distinções com as medidas clássicas de probabilidade e credibilidade.

Teorias da Probabilidade, Possibilidade e Medidas de Incerteza

Nesta seção buscaremos fazer algumas comparações da Lógica Fuzzy na óptica da Teoria da Possibilidade com a Teoria da Probabilidade, fonte

de constantes confusões semânticas. Também procuraremos mostrar que as medidas fuzzy incorporam e generalizam algumas outras medidas de incerteza. Em particular, as medidas de credibilidade de Shafer e as medidas de credibilidade consonantes de Shackle. Isto provê uma justificativa clara ao uso de medidas de possibilidade em sistemas especialistas.

Na Teoria de Probabilidade Clássica, a adição de probabilidades no caso contínuo depende do desenvolvimento de uma teoria de integração cujos limites são livremente intercambiáveis com as integrais (somas infinitas), e isto é dependente de alguma teoria de medida.

Uma medida de probabilidade é uma função π numa σ--álgebra (geralmente considerada como o conjunto-potência de algum conjunto) $P(X)$ em [0,1], satisfazendo as seguintes propriedades:

(MP1) $\pi(X) = 1$,

(MP2) Para toda sequência disjunta de conjuntos A_i em $P(X)$

$$\pi(\bigcup_{i \in \mathbb{N}} A_i) = \sum_{i=1}^{\infty} \pi(A_i)$$

onde $\mathbb{N}$ é o conjunto de números naturais, Claramente, $\pi(\emptyset) = 0$.

Se relaxarmos a condição de aditividade, poderemos definir uma *medida fuzzy* como uma função que satisfaz as seguintes condições :

(MF1) $\pi(\emptyset) = 0$, $\pi(X) = 1$;

(MF2) $\forall A, B \in P(X)$, se $A \subseteq B$ então $\pi(A) \leq \pi(B)$ (monotonicidade)

(MF3) Se $\forall i \in \mathbb{N}, A_i \in P(X)$ e $\{A_i\}$ é uma sequência monótona $\{A_1 \subseteq A_2 \subseteq \ldots \subseteq A_n \subseteq \ldots)$ então

$$\lim_{i\to\infty} \pi(A_i) = \pi(\lim_{i\to\infty} A_i)(\textit{continuidade})$$

Uma medida de probabilidade é, portanto, uma caso particular de medida fuzzy.

Uma *medida de credibilidade* é uma medida em X para a qual vale

(MC1) $\pi(\emptyset) = 0$;

(MC2) $\pi(X) = 1, A \in P(X)$ e $0 \leq \pi(A) \leq 1$;

(MC3) $\forall A_i \in P(X)$ temos

$$\pi(A_1 \cup A_2 \cup \ldots \cup A_n) \geq \sum_{i=1}^{n} \pi(A_i) - \sum_{i<j} \pi(A_i \cap A_j) + \ldots + (-1)^{n+1} \pi(A_1 \cap A_2 \cap \ldots \cap A_n)$$

onde $\pi(A)$ é interpretado como o grau de credibilidade que um elemento pertença a dado conjunto A. Desde que $\pi(A) + \pi(\neg A) \leq 1$, uma falta de credibilidade em $x \in A$ não implica necessariamente numa credibilidade mais forte que $x \in \neg A$. A medida de probabilidade é uma medida de credibilidade, e uma medidade de credibilidade é uma medida fuzzy. Medidas de credibilidade são discutidas por Shafer.

Shackle definiu um caso especial de medida de credibilidade chamada *função de credibilidade consonante* onde $\pi(A)$ é interpretada como sendo o grau com o qual alguém se surpreende ao deparar-se com o fato $x \in A$. As condições são:

(MCC1) $\pi_x(\emptyset) = 0$, $\pi_x(U) = 1$,

(MCC2) $\pi_x(A \cap B) = \min(\pi(A) \cap \pi(B))$.

Uma particularização da função de credibilidade consonante são as *medidas de certeza*, as quais satisfazem

(MCT1) $\exists C \subseteq X$ tal que $\pi(A) = 1$ se $C \subseteq A$ e $\pi(A) = 0$ caso contrário.

Dempster e Shafer introduziram a *medida de plausibilidade* a qual é uma medida que satisfaz

(MPl1) $\pi(\emptyset) = 0, \pi(X) = 1$;

(MPl2) $A \in P(X)$ e $0 \leq \pi(A) \leq 1$;

(MPl3) $\forall A_i \in X$ temos

$$\pi(A_1 \cap A_2 \cap \ldots \cap A_n) \geq \sum_{i=1}^{n} \pi(A_i) - \sum_{i<j} \pi(A_i \cup A_j) + \ldots + (-1)^{n+1} \pi(A_1 \cup A_2 \cup \ldots \cup A_n)$$

Aplicações das medidas de credibilidade em pacotes de Inteligência Artificial são encontradas no MYCIN, EMYCIN e no shell ESE da IBM. Todas são *similares*, mas não *idênticas* às vistas acima. Em particular, elas variam no intervalo [-1,1] ao invés de [0,1]. Elas também não admitem cálculos de distribuições , assim como as apresentadas aqui.

Por último, definimos a *medida de possibilidade* conforme sua apresentação original por Zadeh, como medidas nas quais valem

(PSZ1) $\pi_x(\emptyset) = 0$, $\pi_x(U) = 1$,

(PSZ2) para qualquer família $\{A_i\}$ de subconjuntos de X, $\pi(\bigcup_i A_i) = \sup_i \pi(A_i)$.

A medida de possibilidade pode ser transformada numa medida de credibilidade incluindo-se algumas condições . A medida de possibilidade em X finito é uma medida de plausibilidade, que por sua vez é uma medida fuzzy.

As provas das relações de inclusão de medidas podem ser encontradas,

sem excessão, em (Dubois e Prade, 1980), de modo que é desnecessário transcrevê-las aqui.

Como considerações adicionais, notamos que Dubois e Prade afirmam, por analogia à Lógica Modal, que podemos interpretar a função de credibilidade consonante como sendo a noção dual de *medida de necessidade*. Mais ainda, utilizando-se medidas fuzzy é possivel desenvolver uma teoria de integração (desenvolvida pioneiramente por Sugeno em 1974 (Sugeno, 1974)), possibilidades condicionais e obter resultados análogos fuzzy ao Teorema de Bayes.

2.2.1 Variáveis Linguísticas

Considere a proposição :

"esta amostra é *levemente ácida*"

Ela é uma expressão usual em linguagem natural. Mas qual é o seu *significado* ?

Se consultarmos, por exemplo, em (Gargantini et ali, 1970) encontraremos a seguinte definição :

- Se $pH < 5,0$ então a amostra é *fortemente ácida*
- Se $5,0 \leq pH \leq 5,5$ então a amostra é *ácida*
- Se $5,5 \leq pH \leq 6,0$ então a amostra é *medianamente ácida*
- Se $pH > 6,0$ então a amostra é *levemente ácida*

Mas uma amostra cujo pH for 5.995 será *quase* tão *levemente ácida* quanto uma amostra de $pH = 6.005$, mas sua classificação incidirá na

categoria *ácido*. A implicação desta classificação para efeito de tomada de decisão pode ser catastrófica, pois podemos gastar milhões de dólares em fertilizante para prepararmos um solo *levemente ácido*, próprio para uma cultura desejada, com um solo *quase* tão levemente ácido, um mero problema de linguística.

Porém, haveria outra *possibilidade* de representar estes conceitos de modo mais apropriado ?

Bem, é certo que numa faixa próxima de $pH = 6,5$ o solo será *levemente ácido*, sem nenhuma dúvida, por exemplo, entre 6,25 e 6,75. Mas, nas fronteiras deste conceito, apenas podemos afirmar que cada elemento possui um grau de pertinência cada vez menor ao conjunto em relação ao seu antecessor , de modo que ao alcançar determinados extremos (como abaixo de 5,75 e acima de 7,25) o conceito certamente não se aplica.

Isto nos remete diretamente a alguns conceitos vistos anteriormente, como medida *fuzzy*, distribuição de possibilidade, número fuzzy, etc. Podemos representar a noção de conjunto *levemente ácido* através de uma distribuição de possibilidade (uma vez que seus elementos devem possuir apenas propriedades de mensurabilidade através de medidas fuzzy), com as seguintes características:

- $\pi_{LA}(x) = 1, (\forall x | 6.25 \leq x \leq 6.75)$,
- $\pi_{LA}(x)$ é uma função decrescente entre [5.75,6.25] e entre [6.75,7.25],
- fora dos extremos (abaixo de 5.75 e acima de 7.25) o conceito não se aplica.

Observe que esta noção corresponde *exatamente* a um número fuzzy

$F(5.75, 6.25, 6.75, 7.25)$, que é gerado pela distribuição de possibilidade $\pi_{LA} = \mu_F(x), (\forall x \in U)$, onde LA é o conjunto de objetos que possuem (ou potencialmente possuem) a propriedade de serem *levemente ácidos* (um conjunto fuzzy). Portanto, aos elementos de LA poderemos atribuir a eles o *rótulo linguístico* (nome) *levemente ácido.*

Portanto, uma variável fuzzy oferece uma *restrição flexível* dos valores do domínio que podem ser assumidos por uma variável x quando seu *rótulo linguístico* for u.

Formalmente, uma variável linguística foi definida originalmente por Zadeh da seguinte forma:

Definição Uma *Variável Linguística* é caracterizada por uma quintupla

$$(X, T(X), U, G, M) \tag{2.23}$$

onde X é o nome da variável; U é o Universo de Interesse; T(X) é o conjunto de termos de uma linguagem natural ou artificiais possíveis de serem usados quando da descrição de X; G é a regra sintática usada para gerar os termos de $T(X)$; e M é uma regra semântica que define os significados de $T(X)$. Esta semântica associa cada termo x de $T(X)$ com a variável-base u de acordo com a compatibilidade $\mu_{R_x}(u)$ com o conjunto fuzzy $T(X)$.

Podemos representar outros conceitos difusos, como gradações e outros nomes possíveis no conjunto de acidez, onde cada nome será representado por um conjunto fuzzy $T(X)$ definido pela restrição $R_X(X)$ associada a cada termo de $T(X)$.

Neste contexto, o valor real de x sujeito a T é

$$\text{Se } \alpha \leq t \leq \beta \text{ então } x_i = \mu_{R(x_i)}(t)|T(x_i) \tag{2.24}$$

ou

$$x = \sum_{T(X)} \mu_R(x_i)|T(x_i) \tag{2.25}$$

onde α e β são valores no domínio da escala de acidez [0,14].

Mas o conceito de variável linguística não seria uma redundância aos valores numéricos associados aos conjuntos fuzzy ? Não, os valores linguísticos são *expressões em linguagem natural ou artificial* que descrevem de modo abstrato alguma quantidade, que é então determinada por uma restrição fuzzy num domínio de valores possíveis, que é, por sua vez, um conjunto fuzzy.

2.2.2 Modificadores Linguísticos

Outra importante propriedade linguística existentes entre os nomes (rótulos) associados aos objetos é a capacidade de *modificação linguística.* Por exemplo, consideremos agora o significado da expressão

"esta amostra têm uma acidez *mais ou menos baixa*"

Ora, a aplicação de *mais ou menos* ao nome *acidez baixa* *modifica* seu significado. Será que devemos então ter tabelas para toda a modificação possível de nomes ? Isto é, no mínimo, impraticável. Porém, podemos utilizar as operações de conjuntos fuzzy e as distribuições de possibilidade, provocando assim uma alteração de seu significado.

Por exemplo, consideremos que o conceito *acidez baixa* seja um termo básico de $T(x)$, definido em termos do número fuzzy B que o representa. Então o modificador *mais ou menos* causa uma menor especificidade do nome, aumentando a possibilidade de que outros valores do domínio em que se pode aplicar o nome *acidez baixa* sejam possíveis; considerando os operadores de conjuntos fuzzy vistos anteriormente, um dos operadores possíveis de ser aplicado que cause a modificação desejada no conjunto fuzzy B é o operador de dilatação *dil*:

$$\mu_{\pm \text{baixo}}(x) = dil(\text{baixo}) = \mu^{0.5}_{\text{baixo}}(x)$$

Portanto, através da aplicação do operador apropriado, podemos modificar o significado dos conjuntos que representam os termos básicos de modo a possibilitar a descrição de um espectro muito maior de nomes como:

nome	operador	restrição
muito A	$con(A)$	$\mu_{muito\ A}(x) = \mu^2_A(x)$
$\pm A$	$dil(A)$	$\mu_{\pm A}(x) = \mu^{0.5}_A(x)$
não A	$neg(A)$	$\mu_{não\ A}(x) = 1 - \mu_A(x)$

Também é possível escrever nomes bastante complexos como *razoavelmente, algo, um tipo de*, através da composição dos operadores:

nome	operador	
razoavelmente A	$con(norm(dil(conA))$ e $norm(int(con(A))))$	
algo A	$norm(int(dil(A)$ e $int(dil(nao(A))))$	
$\pm A$	$dil(A)$	$\mu_{\pm A}(x) = \mu^{0.5}_A(x)$
um tipo de A	$norm(int(dil(A)$ e $neg(A)))$	

Convém lembrar que estas operações foram compostas de modo a possuirem a mesma interpretação semântica em termos de ganho ou perda de

pertinência dos elementos do conjunto, mantendo a ordem parcial. Outros operadores podem ser definidos para representar as relações acima descritas, dependendo da semântica que se estiver utilizando.

Uma das confusões que ainda podem ser levadas em consideração é em relação aos significados de *muito, mais, menos, maior, pouco etc.*. por exemplo, *mais e menos* podem ser representados através de translações dos conjuntos fuzzy, da mesma forma que *alto* ou *baixo*, não devendo portanto ser confundidos com o modificador *muito*, que representa um aumento da restrição aos objetos pertencerem a categoria. Estas confusões, embora em maior grau de complexidade, lembram a discussão da semântica apropriada (escolha de conectivos binários) na Lógica Clássica para a descrição de frases linguísticas, não sendo portanto, novidade estarem também presentes aqui.æ

2.2.3 Aproximação Linguística

O problema inverso de se atribuir um conjunto fuzzy a um rótulo linguístico é o de se associar um rótulo linguístico dado um conjunto fuzzy. Este procedimento é chamado de *Aproximação Linguística* (AL) e foi proposto pela primeira vez em 1977 por Mamdani, para a utilização da translação de regras em linguagem natural para aplicações em controle.

Usaremos aqui a definição de aproximação linguística no sentido de Kacprzyk (1986). O procedimento de aproximação linguística consiste em atribuir ao conjunto fuzzy $Y \subseteq U$ um rótulo P^* tal que

$$de(Y, M(P^*)) = \min_{P \in T_x} de(Y, M(P)) \qquad (2.26)$$

onde de(Y,M(P)) denota a distância euclidiana entre os conjuntos fuzzy Y e $M(P)$ e o conjunto $M(P)$ é resultante da aplicação de um modificador linguístico M sobre o conjunto fuzzy P.

O termo P^* na expressão (2.26) será denotado por $AL_x(Y)$. Se houver mais de um termo que satisfiça (2.26), será tomado o mais simples (que contiver menor número de proposições elementares em sua composição).

Convém lembrar que outras medidas de distância (principalmente as medidas de similaridade e compatibilidade) podem ser utilizadas, conforme o caso.æ

2.3 Sintaxe e Semântica da Lógica Fuzzy

A Lógica Fuzzy é a base da argumentação usando conceitos difusos (Zadeh,1973). Esta afirmação geral pode ser refinada de vários modos. Nós consideraremos aqui o caso no qual os argumentos são construidos utilizando-se como base a Teoria de Conjuntos Fuzzy.

2.3.1 Proposições Fuzzy

Por uma *proposição fuzzy* entenderemos uma afirmação da forma "x é A", onde x é o nome de um objeto e A é um conjunto fuzzy no universo U, o qual é uma "restrição flexível" de U de modo a representar o conceito difuso que caracteriza A. Por definição , o valor-verdade numérico $v(A)$ da proposição "x é A" é igual ao grau de pertinência $\mu_A(x)$.

2.3.2 Propriedades das Proposições Fuzzy

Como vimos, podemos calcular o valor numérico das proposições que envolvam proposições fuzzy "x é A" e "x é B" na qual A e B são conjuntos fuzzy dos conjuntos universo (geralmente distintos) U e V. Tal valoração é definida (também geralmente) pelo generalização das fórmulas conhecidas da Lógica multivalente (Zadeh,1988). Daí segue que o valor verdade numérico das proposições compostas "x é A e B","x é A *ou* B" e "x é $\neg A$" são os graus de pertinência do objeto x aos conjuntos difusos $A \cup B$, $A \cap B$ e $\overline{A}$:

$$v(AeB) = v(A) \wedge v(B) \tag{2.27}$$

$$v(AouB) = v(A) \vee v(B) \tag{2.28}$$

$$v(\neg A) = \mathbf{1} - v(A) \tag{2.29}$$

Assim, a proposição "x é A e B" define uma relação binária fuzzy no produto cartesiano dos respectivos conjuntos universos.

A proposição "x é A" também possui um valor verdade linguístico (verdadeira, mais ou menos verdadeira, etc.) que é um número no intervalo [0,1]. Neste caso, o valor verdade da proposição composta (que será denotada aqui por $v(A)$) pode ser obtido através dos seguintes princípios de generalização :

$$\mu_{v(A \text{ e } B)}(t) = \sup_{t=x \wedge y} (\mu_{v(A)}(x) \wedge \mu_{v(B)}(y)), \tag{2.30}$$

$$\mu_{v(A \text{ ou } B)}(t) = \sup_{t=x \vee y} (\mu_{v(A)}(x) \wedge \mu_{v(B)}(y)), \quad (2.31)$$

$$\mu_{v(\neg A)}(t) = \sup_{t=1-x} (\mu_{v(A)}(x)) \quad (2.32)$$

Aqui as variáveis t, x e y tomam valores no intervalo [0,1].a

2.4 Representação da Linguagem Fitosociológica através da Teoria Fuzzy

Esta seção apresenta uma descrição lógica da tipologia fitosociológica. A vaguidão das unidades de vegetação, referenciada por muitos autores, tem sido explicada por meio de ferramentas formais desenvolvidas pela teoria de conjuntos fuzzy.

A tipologia da vegetação tem sido apresentada como variáveis linguísticas do *tipo de comunidade* (*tc*). Um conjunto de suas variáveis linguísticas $\mathcal{T}_{tc}$ contém nomes de sintaxas (como termos primários) e alguns termos compostos.

De acordo com a abordagem linguística proposta, a identificação de comunidades de plantas (fitocoenoses) é baseada em sua composição florística e consiste em atribuir o termo mais apropriado em $\mathcal{T}_{tc}$ a esta fitocoenose. Um procedimento denominado *aproximação linguística* é então utilisado. As diferenças lógicas entre taxa (tipos de plantas) e sintaxa (agrupamento de taxas), e a aplicação da variável *tc* em sistemas especialistas também são discutidas.

2.4.1 Introdução ao Problema Biológico

Este trabalho lida com a usual inflexibilidade dos sistemas fitosociológicos que provém da vaguidão dos nomes de sintaxa. Também é apresentada aqui uma interpretação lógica da tipologia da vegetação usando o cálculo da teoria de conjuntos fuzzy (TCF) introduzido originalmente nas ciências da vegetação por Roberts (1986, 1989) e Feoli e Zucarelo (1988).

A escolha da TCF como base conceitual para a ciência da vegetação aparenta ser bastante apropriada. A TCF é particularmente útil na descrição de sistemas complexos (Zadeh, 1973), e portanto para comunidades de plantas.

Embora o presente trabalho utilize os conceitos da TCF, referimo-nos (ao contrário dos artigos citados) à vaguidão como propriedade da linguagem na qual a vegetação é descrita. Tal abordagem permite análises dos problemas associados com a teoria da vegetação , os quais não podem ser razoavelmente esclarecidos sem que se diga nada sobre a linguagem. Os exemplos incluem: a identificação de comunidades de plantas, predição , a estrutura das generalizações fitosociológicas e, finalmente, o problema dos valores verdade das proposições em fitosociologia. Identificação de comunidades de plantas consiste na seleção de nomes apropriados para uma dada fitocoenose, assume o conhecimento do significado de todos os nomes de sintaxa. Predição é sempre algum tipo de inferência, e portanto ela consiste em derivar uma certa proposição de algum subconjunto dado de outras proposições . O problema da verdade apenas pode ser colocado

de modo razoável quando a relação de certas proposições à realidade extra-linguistica forem levadas em consideração .

Com a reconstrução formal da linguagem fitosociológica podemos iniciar nossa análise da tipologia da vegetação .

2.4.2 Idéias Básicas

Na opinião de muitos fitosociologistas, as unidades de vegetação não são classes (conjuntos nítidos) mas tipos, considerados geralmente como "agrupamentos" no espaço abstrato de vegetação . Intuições consistindo na natureza dos tipos, em particular na sua vaguidão, foram verbalizados por muitos investigadores, como:

- " ... o objetivo da sintaxonomia não é o de proporcionar uma exaustiva divisão em classes distintas, senão a formação de um sistema de referências adequado, permitindo a determinação sistemática da posição da fitocoenose concreta .... O conhecimento dos tipos e de suas características é requisito necessário a identificação própria das formas transacionais ou de casos extremos." (Matuszkiewicz 1981).
- " ... o grupo de espécies diagnosticadas nem sempre é representado em sua plenitude por qualquer amostra simples. De fato, podemos encontrar uma série contínua de amostras com um número decrescente de espécies diagnosticadas, daquelas de maior número, as quais representam o núcleo do sintaxon, até amostras pobremente caracterizáveis nas bordas do sintaxon, as quais não são nítidas." (Moravec 1989).

Não existem características precisas dos aspectos formais da sintaxonomia na literatura, exceto pela afirmação de Moravec de que o procedimento de formação dos sistemas de comunidades de plantas "está em total acordo com a tipologia lógica generalizada por Hempel e Oppenheim" (Moravec 1989). Hempel (1965) escreveu: " ...se o termo T é um tipo extremo, um indivíduo a não pode ser dito T ou não-T; ao invés disto, a pode ser "mais ou menos T". Mas como podemos definir "mais ou menos" de um modo objetivo ?". Lembrando da solução proposta por Hempel e Oppenheim (1936) (veja também Hempel 1965). Um tipo T é definido especificando-se duas relações binárias: "ser mais T que" e ser "tão T quanto". Eles devem satisfazer certas condições formais: a primeira deve ser assimétrica e transitiva, a última deve ser simétrica e transitiva e as duas juntas devem satisfazer para cada par de objetos a e b uma, e somente uma, das seguintes proposições :

1. a é mais T que b,
2. b é mais T que a,
3. a é tão T quanto b.

Todavia, este conceito é apenas uma resposta evasiva às questões relevantes relativas ao significado das expressões da forma "a é mais ou menos T". A vaguidão da frase considerada é, de fato, ignorada por Hempel e Oppenheim: as relações introduzidas por elas são nítidas ! O coração da idéia em questão é o de erradicar as afirmações vagas do discurso científico. Mas a reconstrução da linguagem fitosociológica negligenciando a vaguidão de seus termos básicos pode levar a super-simplificação . Isto

deve-se ao fato de que frases imprecisas são fortemente enraizadas no discurso fitosociológico e possuem papel importante nele, como pode-se mostrar a seguir. A interpretação lógica da tipologia da vegetação que obedece mais fielmente ao espírito da fitosociologia mais do que o conceito de Hempel e Oppenheim será discutida a seguir.

Modificações linguísticas dos nomes da sintaxa

Será útil analisar inicialmente um exemplo não fitosociológico. Consideremos duas sentenças atômicas:

1. *João é careca*
2. *João é* mais ou menos *careca*

Estas proposições possuem a mesma estrutura (são proposições elementares tipo "x é P", onde x é um nome próprio e P um predicado). A palavra vaga "careca" é simplesmente um nome, enquanto a expressão "mais ou menos careca" é um nome composto. A última expressão consiste no nome "careca" e na expressão "mais ou menos" que do ponto de vista gramatical é um *funtor formador de nomes*. Uma expressão é um funtor formador de nomes quando ela se referir a um ou mais substantivo simples de modo a formar um substantivo complexo. O funtor "mais ou menos" modifica o significado do nome "careca" tornando-o mais vago. Retornando aos nomes da linguagem da ciência da vegetação , consideremos as proposições synsistemáticas nas quais fitocoenoses são atribuidas a sintaxas particulares:

1. "a *única comunidade na reserva* é *Quercо-Carpinetum stachyetosum* ..." (Ferchmin)
2. "a maostra foi identificada ...como um fragmento empobrecido de *Carici remotae-Fraxinetum* ..." (Falinski)
3. "...as *comunidades discutida* lembram muito *Saliceum albo-fragilis*" (Polakowski)
4. "*Esta amostra* ...é uma forma transacional entre *Carici-Agrostetum* e *Valeriano-Caricetum flavae*". (Grodzinska)
5. " ...o conjunto de todas as plantas encontradas [na reserva] não permite classificar a amostra como sendo *Ficario-Ulmetum*". (Ferchmin)
6. "*A comunidade* difícil de classificar de alguma forma desenvolve-se *na declividade da montanha Zuchowa*". (Trzcinska-Tacik et alli)

Nomes próprios de *comunidades de plantas* são anotados em destaque; são as contrapartidas ao nome "João" no exemplo anterior. Alguns "elementos adicionais" (sublinhados) fazem o mesmo papel semântico nas sentenças *b-e* do que "mais ou menos" na proposição (ii): eles modificam o significado dos nomes de sintaxa. É evidente que a proposição (1) é uma proposição atômica. Já as proposições 2 – 6 são gramaticalmente mais sofisticadas. De modo a tornar possível a análise formal destas sentenças, podemos assumir que elas possam ser parafraseadas (com a preservação de seu contexto essencial) em sentenças atômicas

$$f \acute{e} T$$

onde f é um nome próprio da fitocoenose, e T denota ou o nome de um sintaxon ou um termo complexo contendo um funtor formador de nome e um ou mais nomes de sintaxa. As transformações são da seguinte forma:

7. f é *Querco-Carpinetum stachyetosum*
8. f é um fragmento empobrecido de *Carici remotae-Fraxinetum*
9. f é aproximadamente *Salicetum albo-fragilis*
10. f está entre *Carici-Agrostetum* e *Valeriano-Caricetum flavae*
11. f não é *Ficario-Ulmetum*
12. f é *indeterminado*

Nas parafrases acima das sentenças 1 – 6 os nomes próprios foram denotados por f, enquanto os funtores formadores de nome foram sublinhados. Na sentença 7 no termo T é simplesmente o nome do sintaxon. Nas proposições 8,9 e 11, T é um termo composto de um modificador lingnístico e o nome de um sintaxon. Na sentença 10, T é um termo construido por um funtor formador de nome e dois nomes de sintaxa. O exemplo 11 mostra que existem comunidades para as quais não podemos atribuir a uma sintaxa específica. A tradução da sentença 12 requer a introdução do nome adicional "indeterminado".

Os exemplos apresentados nesta seção indicam que uma representação lógica adequada da tipologia fitosociológica deve levar em consideração não apenas nomes de sintaxa, mas também os termos compostos que o compõem e os funtores formadores de nome. Na seção seguinte, tanto nomes simples quanto compostos serão traduzidos na linguagem de conjuntos fuzzy.

2.4.3 Pressupostos Formais

Seja F o conjunto de toda a taxa, isto é, da flora investigada na área, e P o conjunto de todas as fitocoenoses que podem ocorrer na mesma área. Seja f a representação do nome próprio de uma fitocoenose. Assume-se que F seja finito e P enumerável. Por simplicidade, suponhamos que cada fitocoenose é caracterizada qualitativamente (isto é, apenas a presença ou ausência das espécies é considerada). Podemos citar dois motivos para a validação desta hipótese: o primeiro é que pela escola de Braun-Blanquet, as comunidades de plantas são reconhecíveis pela sua composição florística; e em segundo, as considerações abaixo podem ser generalizados facilmente nos casos onde a cobertura, frequência ou biomassa de cada espécie for considerada. A composição florística das fitocoenoses será identificada como sendo uma função FC de P no conjunto potência $P(F)$, isto é, o conjunto que contêm todos os subconjuntos de F

$$FC : P \rightarrow P(F)$$

A função FC associa a cada fitocoenose o conjunto de taxa que ocorre em determinada área. Denotemos por A o conjunto de m fitocoenoses descritas (um conjunto de m amostras) que servem como base da síntese sintaxonômica numa dada área. Os elementos de A são simbolizados por $a_1, \ldots, a_m$. É obvio que $A \subset P$. Suponha que n sintaxas (unidades abstratas de vegetação , simbolizadas por $S_1, \ldots, S_n$) foram distinguidas. O conjunto de todos os nomes de sintaxa é denotado por S e seus elementos

por $S_1, \ldots, S_i, \ldots, S_n$ (ou simplesmente S); S_i é o nome do sintaxon S_i. A combinação de espécies características $EsC_i \cup F$, para $i = 1, \ldots, n$, será considerado na definição do i--ésimo sintaxon apresentado abaixo.

Considerações adicionais serão ilustradas através do exemplo seguinte. Assumimos que o conjunto A contenha 20 elementos, isto é, $m = 20$. Suponha que 10 taxas a,...,j tenham sido encontradas nestas fitocoenoses. A composição floristica de cada amostra é exibida na tabela 1; Por exemplo, $FC(a_1) =$ { a,b,f }. Seja $n = 3$, $EsC_1 =$ { a,b,c } , $EsC_2 =$ { d,e,f,g } , $EsC_3 =$ { d,e,h,i }.

Tabela 1. Tabela fitosociológica estruturada a partir de 20 relevés artificiais (a1,...,a20), nos quais 10 taxa (a,...,j) foram encontrados

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	a16	a17	a18	a19	a20	
a	1		1	1	1	1	1	1			1			1							
b	1	1	1		1		1	1				1			1						
c		1	1	1	1	1		1	1							1					
d					1				1		1	1	1	1	1	1	1		1	1	
e		1					1		1	1	1	1		1	1	1	1	1	1	1	æ
f	1								1	1		1	1	1	1				1		
g				1				1	1	1	1	1	1	1						1	
h										1					1	1	1	1		1	
i						1									1	1		1	1	1	
j			1				1				1					1			1		

Variável Linguística *tipo de comunidade*

Intuições relacionadas a subjetividade da sintaxa e da suscetibilidade dos nomes das comunidades de plantas aos modificadores linguisticos devem ser formalmente expressos através da variável linguistica referente a tipologia fitosociológica.

Variáveis linguisticas, conforme vimos anteriormente, são uma quintupla (conforme Zadeh 1975)

$$(v, T_t c, A, G, M) \tag{2.33}$$

onde v é seu nome, $\mathcal{T}_{tc}$ é o conjunto de variáveis linguisticas, A é o conjunto de todas as amostras, G é a regra sintática nos quais os termos T em $\mathcal{T}_{tc}$ são gerados, e M é a regra semântica que atribui um conjunto fuzzy a cada termo em $\mathcal{T}_{tc}$.

O conjunto $\mathcal{S}$ é o único subconjunto próprio de $\mathcal{T}_{tc}$. O último deve também incluir termos compostos de nome de sintaxa e de certas expressões, consideradas como modificadores linguisticos. Os referidos funtores modificam o sentido dos nomes de sintaxa. Por exemplo, a expressão "um fragmento empobrecido de" em conexão com o nome S_i modifica o sentido deste. Podemos levar em consideração que o significado da expressão "um fragmento empobrecido de" como um conjunto fuzzy resultante de uma operação realizada num conjunto fuzzy que representa o sentido de S_i. O sentido de qualquer termo S_i em $\mathcal{S}$ é o sintaxon S_i, um conjunto fuzzy de A definido como segue:

$$\mathrm{S}_i = norm((Z_i)^a) \tag{2.34}$$

$$\mu_{Z_i}(a_j) = I(EsC_i, FC(a_j))$$

onde $i = 1, \ldots, n$; $j = 1, \ldots, m$, $norm((Z_i)^a)$ é o conjunto normalizado da a–ésima potência de $Z_i \subseteq A$, $(a \geq 1)$, e $I(EsC_i, FC(a_j))$ é o grau de inclusão do conjunto nítido EsC_i em $FC(a_j)$[2].

Vamos definir o grau de inclusão I como

$$I(X,Y) = \frac{|X \cap Y|}{|X|}$$

[2]Para definição dos operadores fuzzy utilizados, consulte as seções anteriores deste capítulo

onde $|.|$ denota a cardinalidade, e X e Y são conjuntos fuzzy.

O grau de pertinência da fitocoenose a_j no sintaxon S_i cresce de acordo com o número de espécies diagnosticadas (isto é, espécies pertencendo a EsC_i) encontradas nela. O grau de pertinência de a_j em S_i é 1 se, e somente se, $EsC_i \subseteq FC(a_j)$. O parâmetro a na definição (2.34) determina o grau de subjetividade (segundo Kapriczyk 1986) de S_i: Quanto maior a, menos fuzzy é S_i. Em particular, quando $a \to \infty$, então S_i é um conjunto nítido.

As regras sintáticas (G)para a geração de elementos de $\mathcal{T}_{tc}$ e da regra semântica correspondente (M) é apresentada na tabela 2.

Tabela 2. Regra sintática e correspondente semântico da variável linguística *tipo de comunidade*. s_i e S_j são nomes de sintaxa S_i e S_j; $i, j = 1, \ldots, n$.

Regra Sintática (G)	Regra Semântica (M)
$S_i \in \mathcal{T}_{tc}$	$M(S_i)=S_i$
"um fragmento empobrecido de $S_i \in \mathcal{T}_{tc}$"	$M($"um fragmento empobrecido de S_i"$)=(S_i)^b, b>1$
"aproximadamente S_i"$\in \mathcal{T}_{tc}$	$M($"aproximadamente S_i"$)=(S_i)^c, c<1$
"não S_i"$\in \mathcal{T}_{tc}$	$M($não $S_i)=\neg S_i$
"forma transacional entre S_i e S_j"$\in \mathcal{T}_{tc}, i \neq j$	$M($"forma transacional entre S_i e S_j"$)=(S_i \cup S_j)^d, i \neq j, d>1$
"indeterminado"$\in \mathcal{T}_{tc}$	$M($"indeterminado"$)=0/a_k, k=1,\ldots,m$

A variável linguística *tipo de comunidade* possui um número finito de termos. O conjunto $\mathcal{T}_{tc}$ consiste em todos os elementos de $\mathcal{S}$ e os termos não primários (compostos) compostos de modificadores linguísticos e um número adequado de termos básicos (proposições elementares). O termo "indeterminado" completa a lista de $\mathcal{T}_{tc}$.

O significado do nome S_i é o sintaxon S_i (conjunto fuzzy) S_i. O significado do nome composto "uma amostra empobrecida de S_i", é a $b-$ésima potência do conjunto S_i: $(S_i)^b$, $b > 1$. Uma diminuição do grau de pertinência de cada fitocoenose a_j (com excessão daquelas cujo grau seja

nítido (=1)) é o reflexo formal do empobrecido florístico geral. Este não é o caso em que uma fitocoenose é aproximadamente S_i: algumas espécies representando outras sintaxas são encontradas nela. Um aumento do grau de pertinência de cada fitocoenose a_j, pela aproximação dos valores da função de pertinência de S_i à $c-$ésima potência ($c < 1$), equivale a referida diferença entre o significado do nome primário S_i e S'_j é a $d-$ésima potência da união $S_i \cup S_j$, onde $d > 1$. O nome "indeterminado" é o rótulo linguístico do conjunto vazio.

O conjunto T_{tc} definido acima contem um número finito de elementos. Embora ele possa não se referir à uma variedade completa de nomes de comunidade aplicadas na prática fitosociológica, ele contêm as constituições mais geralmente encontradas. Este conjunto pode ser facilmente estendido pela adição de nomes da forma "intermediária entre S_i e S_j e ...e S_k" etc. Um conjunto infinito de termos da variável linguística *tipo de comunidade* pode ser gerado por uma gramática livre de contexto (Zadeh 1975). As regras semânticas apresentadas na tabela 2 devem ser pensadas como sendo uma primeira aproximação do significado real das expressões examinadas.

As regras descritas serão aplicadas aos dados da tabela 1.

Tabela 3. Conjuntos Fuzzy construídos a partir dos dados da tabela 1 e suas variações linguísticas.

	1	2	3	4	5	6	7	8	9	10
$S1$	0.60	0.60	1.00	0.60	1.00	0.60	0.60	1.00	0.25	0.00
$S2$	0.18	0.18	0.00	0.18	0.18	0.18	0.18	0.18	1.00	0.70
$S3$	0.00	0.18	0.00	0.00	0.18	0.18	0.18	0.00	0.42	0.42
$empS1$	0.13	0.13	1.00	0.13	1.00	0.13	0.13	1.00	0.00	0.00
$empS2$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.24
$empS3$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.03
$\mp S1$	0.77	0.77	1.00	0.77	1.00	0.77	0.77	1.00	0.50	0.00
$\mp S2$	0.42	0.42	0.00	0.42	0.42	0.00	0.42	0.42	1.00	0.84
$\mp S3$	0.00	0.42	0.00	0.00	0.42	0.42	0.42	0.00	0.65	0.65
$naoS1$	0.40	0.40	0.00	0.40	0.00	0.40	0.40	0.00	0.75	1.00
$naoS2$	0.82	0.82	1.00	0.82	0.82	1.00	0.82	0.82	0.00	0.30
$naoS3$	1.00	0.82	1.00	1.00	0.82	0.82	0.82	1.00	0.58	0.58
$S1\&S2$	0.36	0.36	1.00	0.36	1.00	0.36	0.36	1.00	1.00	0.49
$S1\&S3$	0.36	0.36	1.00	0.36	1.00	0.36	0.36	1.00	0.18	0.18
$S2\&S3$	0.03	0.03	0.00	0.03	0.03	0.03	0.03	0.03	1.00	0.49
$indet$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	11	12	13	14	15	16	17	18	19	20
$S1$	0.25	0.25	0.00	0.25	0.25	0.25	0.00	0.00	0.00	0.00
$S2$	0.70	1.00	0.70	1.00	0.70	0.42	0.42	0.18	0.70	0.70
$S3$	0.42	0.42	0.18	0.42	1.00	1.00	0.70	0.70	0.70	1.00
$empS1$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$empS2$	0.24	1.00	0.24	1.00	0.24	0.03	0.03	0.00	0.24	0.24
$empS3$	0.03	0.03	0.00	0.03	1.00	1.00	0.24	0.24	0.24	1.00
$\mp S1$	0.50	0.50	0.00	0.50	0.50	0.50	0.00	0.00	0.00	0.00
$\mp S2$	0.84	1.00	0.84	1.00	0.84	0.65	0.65	0.42	0.84	0.84
$\mp S3$	0.65	0.65	0.42	0.65	1.00	1.00	0.84	0.84	0.84	1.00
$naoS1$	0.75	0.75	1.00	0.75	0.75	0.75	1.00	1.00	1.00	1.00
$naoS2$	0.30	0.00	0.30	0.00	0.30	0.58	0.58	0.82	0.30	0.30
$naoS3$	0.58	0.58	0.82	0.58	0.00	0.00	0.30	0.30	0.30	0.00
$S1\&S2$	0.49	1.00	0.49	1.00	0.49	0.18	0.18	0.03	0.49	0.49
$S1\&S3$	0.18	0.18	0.03	0.18	1.00	1.00	0.49	0.49	0.49	1.00
$S2\&S3$	0.49	1.00	0.49	1.00	1.00	1.00	0.49	0.49	0.49	1.00
$indet$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Assumiremos que os parâmetros (veja fórmula (2.34) e tabela 2) são $a = 1.25$, $b = 4$, $c = 0.5$ e $d = 2$. O conjunto fuzzy que são os significados dos termos linguisticos em $\mathcal{T}_{tc}$ são esclarecidos em seguida. A título de ilustração , consideremos o nome "intermediário entre S_1 e S_3". De acordo com as regras semânticas apresentadas na tabela 2, o significado deste nome é a d–ésima potência da união $S_1 \cup S_3$:

$$S(\text{"formação intermediária entre} S_1 \text{ e } S_3\text{"}) = (S_1 \cup S_3)^d.$$

Para calcular o conjunto fuzzy S_1, o conjunto $Z_1 \subseteq A$ é determinado (veja a fórmula (2.34)). O grau de pertinência de a_j, $(j = 1, \ldots, 20)$ em Z_1 é igual à $I(EsC_1, FC(a_j))$. Por exemplo,

$$\mu_{Z_1}(a_1) = I(EsC_1, FC(a_1))$$

$$= \frac{|\{a,b,c\} \cup \{a,b,f\}|}{|\{a,b,c\}|}$$
$$= \frac{|\{a,b\}|}{|\{a,b,c\}|} = \frac{2}{3}$$
$$\mu_{Z_1}(a_3) = I(EsC_1, FC(a_3)) = 1$$

Cada valor $\mu_{Z_1}(a_j)$, $(j = 1, \ldots, 20)$, é agora elevado a a–ésima potência de modo a obter o conjunto $(Z_1)^a$. Tomando $a = 1.25$, teremos:

$$\mu_{Z_1{}^a}(a_1) = (\frac{2}{3})^{1.25} = 0.6$$
$$\mu_{Z_1{}^a}(a_3) = 1^{1.25} = 1$$

Agora, o peso de $(Z_1)^a$ é calculado. Como $\mu_{Z_1{}^a}(a_3) = \mu_{Z_1{}^a}(a_5) = \mu_{Z_1{}^a}(a_8) = 1$, temos que $peso((Z_1)^a) = 1$. Isto nos dá

$$S_1 = norm((Z_1)^a) = (Z_1)^a$$

Assim, $\mu_{S_1}(a_1) = 0.6$ e $\mu_{S_1}(a_3) = 1$.

O conjunto S_3 é determinado do mesmo modo. O próximo passo é o cálculo de $S_1 \cup S_3$, como segue:

$$\mu_{S_1 \cup S_3}(a_1) = \max\{\mu_{S_1}(a_1) \cup \mu_{S_3}(a_1)\} = \max\{0.6, 0\} = 0.6$$
$$\mu_{S_1 \cup S_3}(a_3) = \max\{\mu_{S_1}(a_3) \cup \mu_{S_3}(a_3)\} = \max\{1, 0\} = 1$$

Finalmente, cada valor $\mu_{S_1 \cup S_3}(a_j)$ é elevado á d–ésima potência. Fazendo $d = 2$, temos:

$$\mu_{S_1 \cup S_3}{}^{d}(a_1) = (0.6)^2 = 0.36$$

$$\mu_{S_1 \cup S_3}{}^{d}(a_3) = (1^2) = 1$$

2.4.4 Identificação de comunidades

À luz das observações anteriores, apenas os elementos do conjunto A pertencem ao sintaxon S_i num grau variando no intervalo $[0,1]$. Consideremos agora o seguinte problema: Em que pode a identificação dos elementos da diferença $P - A$ consistir, quando nenhum deles pode ser dito pertencente a qualquer sintaxon? O objetivo da sintaxonomia, em concordância com a opinião de Matuzkiewicz citada anteriormente, é que ela não é uma classificação exaustiva do conjunto P, mas sim a formação de um sistema de referências adequado para o estabelecimento de uma posição sistemática de fitocoenoses concretas. O conjunto $\mathcal{T}_{tc}$ é o sistema de referência em questão. A identificação de uma fitocoenose como pertencente à P (em particular, o conjunto A é um subconjunto de P) é verdadeiramente a atribuição a uma comunidade de exatamente um único nome de $\mathcal{T}_{tc}$. À função que associa um nome a toda fitocoenose pode ser denotada por DET:

$$DET : P \rightarrow \mathcal{T}_{tc}.$$

A definição da função DET deve ser construida de modo a estar de acordo com a noção intuitiva com a qual o cientista da vegetação associa naturalmente com o que vem a ser o procedimento de identificação . O

que desejamos com esta definição é encontrar algum procedimento formal que escolha de fato o nome de fitocoenose apropriado, dependendo de sua composição florística e cada comunidade de plantas. Para definir a função DET, será necessário introduzir um conjunto fuzzy auxiliar $Z \subseteq A$ o qual possa ser calculado para qualquer fitocoenose p de composição florística $FC(p)$:

$$\mu_Z(a_j) = SJ(FC(p), FC(a_j)),$$

onde $j = 1, \ldots, m$, e $SJ(FC(p), FC(a_j)$ denota o coeficiente de similaridade de Jaccard (Goodall 1973) entre o conjunto nítido $FC(p)$ e $FC(a_j)$. O conjunto Z é, intuitivamente falando, um reflexo da "concordância florística" entre a fitocoenose p e cada elemento de A. Se o conjunto Z for idêntico a alguns dos conjuntos que representam o significado do termo em $\mathcal{T}_{tc}$, então podemos aceitá-lo como sendo o nome (rótulo linguístico) de p. Mas, geralmente não há tal coincidência. Assim sendo, devemos buscar o termo em $\mathcal{T}_{tc}$ cujo significante possua a menor distância em $norm(Z)$. Este procedimento é chamado de *aproximação linguística*. Podemos agora definir a função DET de modo formal, como sendo

$$DET(p) = AL_{tc}(norm(Z)), \forall p \in P \tag{2.35}$$

onde AL representa a aproximação linguística e $norm(Z)$ é a normalização do conjunto Z calculado para a fitocoenose p. Um modo de verificar a definição (2.35) acima é observar como ela trabalha. Vamos utilizar um exemplo previamente discutido (veja tabela 4).

Tabela 4. Nomes atribuidos a fitocoenoses de diversas composições florísticas.

$FC(p)$	$DET(p)$	$FC(p)$	$DET(p)$
$\{a,b,c\}$	S_1	$\{f,g\}$	$imp.S_2$
$\{d,e,f,g\}$	S_2	$\{i\}$	$imp.S_3$
$\{d,e,h,i\}$	S_3	$\{j\}$	$indet.$
$\{a,b,f\}$	S_1	$\{a,c,f,h,j\}$	$\pm S_1$
$\{c,d,e,f,g\}$	S_2	$\{a,b,d,e,f,g,i\}$	$\pm S_2$
$\{b,d,e,f,h,i\}$	S_3	$\{b,d,e,f,h,j\}$	$\pm S_3$
$\{a,b,j\}$	S_1	$\{a,b,d,e,f,g,i\}$	$S_1\&S_2$
$\{b,e,f,g,j\}$	S_2	$\{b,d,e,h,i\}$	$S_1\&S_3$
$\{a,d,h,i\}$	S_3	$\{d,e,f,g,h,i\}$	$S_2\&S_3$

A composições de espécies presentes a uma comunidade EsC, tipicamente presentes (como as da tabela 1, por exemplo) recebem, de acordo com as expectativas, nomes primitivos ("puros"). Comunidades intermediárias ou "empobrecidas" recebem nomes compostos apropriados. Portanto, a definição da função DET parece conseguir capturar a idéia de identificação satisfatoriamente.

2.4.5 Discussão

Quais são as maiores diferenças do ponto de vista lógico entre taxa e sintaxa ? Primeiramente, taxa é usualmente referida como sendo conjuntos nítidos (Gregg 1954; Buck e Hull 1966), aonde sintaxa pode ser referida como sendo um conjunto fuzzy. Isto é reforçado por considerações de ordem linguística e ontológica. Os nomes de sintaxa são conceitos vagos e geralmente utilizam-se de modificadores linguísticos, enquanto taxa não. Um fitosociologista pode referir-se à "uma amostra empobrecida de *Carici remotae-Fraxinetum*", etc., mas expressões análogas que contenham nomes de taxa não fazem sentido. Estas diferenças linguísticas possuem

uma causa mais profunda ainda: O espaço abstrato de vegetações é mais homogêneo que o espaço de características idiotaxonômicas (Whittaker 1962; Matuszkiewics 1981; Mirkin 1989). O grau de nebulosidade da sintaxa depende da proporção entre continuidade e descontinuidade observada na natureza: quanto maior a descontinuidade, melhor são separadas as unidades de vegetação (Whittaker 1962; Goodall 1963; Westhoff e van der Maarel 1973; van der Maarel 1975; Moravec 1989).

Segundo, taxa são conjuntos potencialmente infinitos (Buck e Hull 1966), enquanto sintaxa são conjuntos finitos. Um taxon é determinado pelas especificações das características que o definem. Um dado individuo reside em um taxon ou fora dele, dependendo da satisfação ou falta das características de definição . Assim, a identificação de um organismo não passa da atribuição de uma classe particular (conjunto nítido) a um organismo num sistema taxonômico. A situação na ciência da vegetação é mais complicada. De acordo com a abordagem acima, a cardinalidade de um sintaxon S, definida como $\sum_{j=1}^{m} \mu_S(a_j)$ (Zadeh 1983), não é maior do que o número de fitocoenoses que possam servir como base da síntese sintaxonômica. Portanto, toda comunidade de plantas que não pertence ao conjunto A, isto é, toda amostra que não puder ser levada em consideração durante a fase sintética da pesquisa, não poderá ser dita pertencente a qualquer sintaxon. Embora toda comunidade de plantas possa ser sintaxonomicamente interpretada (Matuszkiewickz 1981), ela não pode ser subsumida sobre qualquer sintaxon. Podemos observar agora que a tipologia da vegetação pode ser imaginda como sendo um

sistema abstrato bastante complexo com suas próprias regras semânticas e sintáticas nas quais a sintaxa e seus nomes servem apenas como pontos de referência. O estabelecimento da correspondência entre nomes de sintaxa e sintaxa propriamente ditas, as quais podem ser tratadas como conjuntos fuzzy finitos, permite o "cálculo" do nome de fitocoenose em P mais apropriado. O principal papel desta maquinaria lógica é representado pela função DET que atribui a cada elemento de P um nome do conjunto de termos da variável *tipo de comunidade*.

Conjuntos fuzzy os quais são significados de proposições simples e complexas sobre nomes de comunidades de plantas são definidas neste trabalho referentes às combinações de espécies características. É irrelevante como uma EsC particular foi obtida, uma vez que o estágio indutivo da fitosociologia não é o objeto deste trabalho. É obvio que podemos separar a sintaxa melhor ou pior de acordo com a seleção de EsC; apesar disto, a interpretação lógica ainda será a mesma.

Uma certa discussão foi conscientemente omitida neste trabalho. Syntaxons são usualmente definidos como sendo fitocoenon (unidades fitosociológicas abstratas sem posição fixada) com a categoria fitotaxonomica associada a ele (Westhoff e van der Maarel 1973; Mirkin et al. 1989). Mas claramente tem sido de distinguir dentro dos limites deste trabalho estes conceitos de modo detalhado e preciso, desde que explicações sobre a estrutura lógica da hierarquia fitotaxonômica fazem-se necessárias. Este assunto complexo deve ser discutido separadamente. Notemos apenas que a Teoria de Conjuntos Fuzzy nos fornece um aparato formal

adequado à descrição da complexidade da tipologia fitosociológica. Por exemplo: o fato que "as unidades de vegetação são relacionadas umas às outras num reticulado ao invés de sistemas hierárquicos" (van der Maarel 1989) podem ser expressos de modo preciso através da concepção teórica forma dos conjuntos fuzzy de graus de inclusão.

Os resultados aqui apresentados serão utilizados na construção de sistemas especialistas, com a tradução da incerteza do conhecimento fitosociológico em linguagens compreensíveis pelo computador (Buchanan et al. 1983; Noble 1978; Blinder 1993a). A plataforma entre a intuição do fitosociologista e o computador pode ser justamente a Teoria de Conjuntos Fuzzy (Zadeh 1978; Zadeh 1983). Se um sistema especialista deve lidar com qualquer conhecimento referente a sintaxa de uma área, ele deve "saber" não apenas todos os nomes primários e nomes linguisticamente modificados, mas também seu significado. Em outras palavras, o sistema deve ser capaz de "compreender" as diferenças semânticas sutis entre *"Ficario-Ulmentum"* e *"aproximadamente Ficario-Ulmentum"* e assim por diante. Tal capacidade de "compreensão" de conceitos imprecisos através do computador é possível através da Lógica Fuzzy (Maranca 1993). Como foi mostrado neste trabalho, tanto as regras semânticas quanto sintáticas da fitosociologia podem ser facilmente representadas de um modo formal utilizando-se o conceito de variável linguística. Os exemplos mostrados exibem que a semântica proposta coincide com nossa intuição . Detalhes técnicos de como montar bases de conhecimento utilizando variáveis linguísticas podem ser encontrados em (Zadeh 1978;

Blinder 1993b).

As bases conceituais propostas neste trabalho apresentam uma perspectiva diferente de muitas abordagens teóricas em ciência da vegetação , especialmente àquelas pertencentes à generalização e predição fitosociológicas. Estes problemas serão considerados nas seções seguintes.æ

Capítulo 3

Inferência Fuzzy

3.1 Relação Fuzzy

Uma *relação fuzzy* de um conjunto fuzzy X em um conjunto fuzzy Y é definida como um subconjunto do produto cartesiano $X \times Y$, e é caracterizada pela função de pertinência bivalente $\mu_R(x, y)$, expressa por

$$R(x,y) \triangleq \int_{X \times Y} \mu_R(x,y) \mid (x,y) \; onde \; x \in X \; e \; y \in Y$$

Mais geralmente, uma relação fuzzy n-ária R_n é um subconjunto fuzzy de $X_1 \times X_2 \times \ldots \times X_n$ denotada por

$$R \triangleq \int_{X_1 \times X_2 \times \ldots \times X_n} \mu_R(x_1, x_2, \ldots, x_n) \mid (x_1, x_2, \ldots, x_n) \; onde \; x_i \in X_i, i = 1, \ldots, n$$

Exemplo: Sejam X ={Abel,Caim } e Y ={Paulo,Pedro}. A relação binária de *semelhança* entre os membros de X e Y pode ser expressa por

$$S = 0,8|(Abel, Paulo) + 0,6|(Abel, Pedro) + 0,2|(Caim, Paulo) + 0,9|(Caim, Pedro)$$

Alternativamente, podemos expressar uma relação fuzzy de modo matricial, como:

$$
\begin{array}{c} \\ \text{Paulo} \\ \text{Pedro} \end{array}
\begin{array}{c} R \\ \begin{array}{cc} \text{Abel} & \text{Caim} \end{array} \\ \begin{bmatrix} 0.8 & 0.6 \\ 0.2 & 0.9 \end{bmatrix} \end{array}
$$

onde o (i,j)–ésimo elemento é o valor de $\mu_R(x,y)$.

Relação Fuzzy Composta

Se R é uma relação fuzzy de X em Y e S é uma relação fuzzy de Y em Z, então a composição entre R e S será denotada por $R \circ S$ e definida como sendo

$$\mu_{R \circ S}(x,z) \triangleq \int_{X \times Z} \max_y (\min_{x \in X, z \in Z} (\mu_R(x,y), \mu_S(y,z)))|(x,z)$$

Se os domínios das variáveis x, y e z de $R \circ S$ forem finitos, então uma das relações possíveis de serem aplicadas é a *composição max-min*[1] das matrizes de relação R e S.

Exemplo:

$$
\begin{array}{ccccc} R & & S & & R \circ S \\ \begin{bmatrix} 0.3 & 0.8 \\ 0.6 & 0.9 \end{bmatrix} & o & \begin{bmatrix} 0.5 & 0.9 \\ 0.4 & 1.0 \end{bmatrix} & = & \begin{bmatrix} 0.4 & 0.8 \\ 0.5 & 0.9 \end{bmatrix} \end{array}
$$

Inferência Composicional

Um caso particular da inferência fuzzy é a *Inferência Composicional*. Esta ocorre quando tomamos um conjunto fuzzy X do domínio U, e uma relação fuzzy R, definida de $U \times V$, onde X terá o papel de uma relação fuzzy unária.

[1] No caso da cardinalidade de $R \circ S$ for infinita, então a relação é chamada de *composição sup-min*.

Exemplo: Se $X = 0,2|1 + 1,0|2 + 0,3|3$ e

$$R = \begin{array}{c} \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0,8 & 0,9 & 0.2 \\ 0,6 & 1,0 & 0,4 \\ 0,2 & 0,8 & 1,0 \end{bmatrix} \end{array}$$

então $Y = X\ o\ R = [0,2\ 1,0\ 0,3] o \begin{bmatrix} 0,8 & 0,9 & 0,2 \\ 0,6 & 1,0 & 0,4 \\ 0,5 & 0,8 & 1,0 \end{bmatrix}$

portanto $Y = [0,6\ 1,0\ 0,4] = 0,6|4 + 1,0|5 + 0,4|6$

3.2 Inferência Fuzzy

3.2.1 Implicação Fuzzy

O valor numérico da proposição "Se A então B" (implicação fuzzy) na qual A e B são conjuntos fuzzy dos conjuntos universo (geralmente distintos) U e V também pode ser definida através da generalização das fórmulas conhecidas de inferência da Lógica multivalente (Zadeh, 1988):

$$v(A \Rightarrow B) = (1 - v(A)) \vee v(B), \tag{3.1}$$

$$v(A \Rightarrow B) = (1 - v(A)) \vee (v(A) \wedge v(B)), \tag{3.2}$$

$$v(A \Rightarrow B) = 1 \wedge (1 - v(A) + v(B)) \tag{3.3}$$

Os conjuntos fuzzy cujas funções de pertinência são descritas pelas fórmulas nestas relações são relações fuzzy binárias em $U \times V$ que são respectivamente iguais à

$$R_b = (\neg A \times V) \cup (U \times B), \tag{3.4}$$

$$R_m = (\neg A \times V) \cup (A \times B), \tag{3.5}$$

$$R_a = (\neg A \times V) \oplus (U \times B) \tag{3.6}$$

onde $\oplus$ é a soma-limitada.

Assim, a proposição "Se A então B" define uma relação binária fuzzy no produto cartesiano dos respectivos conjuntos universais.

A proposição "x é A" também possui um valor verdade linguístico (verdadeira, mais ou menos verdadeira, etc.) que é um número no intervalo [0,1]. Neste caso, o valor verdade da proposição composta (que será denotada aqui por $v(A)$) pode ser obtido através dos seguintes princípios de generalização :

$$\mu_{v(A \text{ e } B)}(t) = \sup_{t=x \wedge y} (\mu_{v(A)}(x) \wedge \mu_{v(B)}(y)), \tag{3.7}$$

$$\mu_{v(A \text{ ou } B)}(t) = \sup_{t=x \vee y} (\mu_{v(A)}(x) \wedge \mu_{v(B)}(y)), \tag{3.8}$$

$$\mu_{v(\neg A)}(t) = \sup_{t=1-x} (\mu_{v(A)}(x)), \tag{3.9}$$

$$\mu_{v(A \Rightarrow B)}(t) = \sup_{t=1 \wedge (t-x+y)} (\mu_{v(A)}(x) \wedge \mu_{v(B)}(y)). \tag{3.10}$$

Aqui as variáveis t, x e y tomam valores no segmento [0,1].

3.2.2 Regras de Inferência

O cálculo lógico fuzzy é construído de proposições fuzzy compostas de expressões construídas a partir de regras de inferência. O conjunto inicial de premissas fuzzy é determinado a partir dos modelos concretos.

Algumas das principais regras de inferência serão exibidas nesta seção .

A *regra composicional* é uma das principais regras da Lógica Fuzzy. Esta lei opera de acordo com o esquema

$$\frac{x \text{ é } A' \text{ , } (x,y) \text{ é } R}{y \text{ é } B'} \tag{3.11}$$

onde A,B e R são conjuntos fuzzy em U, V e $U \times V$, respectivamente. A expressão funcional para a lei de composição é dada pela equação

$$\mu_{B'}(y) = \bigvee_{u \in U} (\mu_{A'}(x) \wedge \mu_R(x,y)). \tag{3.12}$$

É claro que esta fórmula é um caso particular da fórmula que define a relação de composição binária de dois conjuntos fuzzy quando temos

$$B' = A' \circ R. \tag{3.13}$$

Um caso especial da lei de composição provê a regra de inferência chamada de *Modus Ponens Generalizado (GMP)*:

$$\frac{x \text{ é } A' \text{ , Se } x \text{ é } A \text{ então } y \text{ é } B}{y \text{ é } B'} \tag{3.14}$$

e, obviamente, no caso mais particular em que $A' = A$ e a proposição "Se A então B" defina alguma relação fuzzy $R = R_m = (\neg A \times V) \cup (A \times B)$, a lei de composição coincide com a própria lei de *Modus Ponens*:

$$\frac{x \text{ é } A \text{ , Se } x \text{ é } A \text{ então } y \text{ é } B}{y \text{ é } B} \tag{3.15}$$

Devemos notar aqui o seguinte fato: O resultado da derivação utilizando *Modus Ponens Generalizado* depende do método que define a relação

fuzzy R correspondente à implicação $A \Rightarrow B$. De modo a tentar reduzir esta dependencia, Zadeh introduziu posteriormente alguns outros esquemas de inferência, como a *lei da inclinação* :

$$\frac{x \text{ é } A}{x \text{ é } A' \ : \ A \subseteq A'} \tag{3.16}$$

onde A e A' são conjuntos fuzzy do mesmo conjunto universo U e suas funções de pertinência satisfazem a condição $(\forall x \in U)[\mu_A(x) \leq \mu_{A'}(x)]$ e a *lei da conjunção* :

$$\frac{x \text{ é } A, \ x \text{ é } B}{x \text{ é } A \text{ e } B} \tag{3.17}$$

e, obviamente, $\mu_{AeB}(x) = \mu_{A \cup B}(x) = \mu_A(x) \wedge \mu_B$.

Podemos observar na literatura sobre conjuntos fuzzy que o cálculo lógico fuzzy não é uma disciplina de investigação independente. Exceções a esta regra são feitas em certos trabalhos que consideram problemas particulares. Geralmente, a Lógica Fuzzy é utilizada em aplicações . Assim, as leis de inferência aqui mostradas foram propostas por Zadeh em conexão com o desenvolvimento de uma teoria de inferência para sistemas especialistas.

3.3 Aplicação : Generalização e Predição em Fitosociologia

Nesta seção trataremos de realizar uma análise das propriedades da linguagem fitosociológica, as quais são correlatas com a vaguidão de alguns conceitos em ciência da vegetação . A tradução de proposições sinsistemáticas elementares em distribuições de possibilidade são propostas. Relações inferenciais entre sentenças mencionadas usando o conceito de

inclusão semântica são referenciadas. Sentenças descritivas de requisitos de habitat de sintaxa são de fato condicionais implícitos; suas paráfrases são dadas na mesma forma semântica que a implicação lógica. Desde que estas sentenças incluem predicados fuzzy, seus significados são relações fuzzy. Estas constituem bases para a predição em termos de regras composicionais de inferência.

3.3.1 Introdução ao Problema Biológico

A linha comum que relaciona esta seção e as anteriores é a vaguidão como propriedade intrínseca da linguagem fitosociológica. Nas seções anteriores foi argumentado que a tipologia da vegetação pode ser representada através de uma variável linguística (*tipo de comunidade (tc)*). De acordo com a abordagem fuzzy proposta, o conjunto de valores linguísticos da variável ($\mathcal{T}_{tc}$) contém como termos os nomes de sintaxa elementares e compostos, os quais são construídos a partir de nomes básicos com o auxílio de modificadores linguísticos, tais como "um empobrecido fragmento de ...","aproximadamente ...", etc. O significado de cada termo T em $\mathcal{T}_{tc}$ é um conjunto fuzzy em A: $S(T) \subseteq A$; onde A é o conjunto de todas as fitocoenoses descritas, as quais servem como base da síntese fitotaxonômica numa dada área.

O conhecimento sobre a vegetação é dado em linguagem natural. Sentenças que descrevem a cobertura por plantas contêm dezenas de termos vagos. Como será visto em seguida, não apenas os nomes de sintaxa são fuzzy, mas também, por exemplo, alguns termos descritores de fatores de habitat. Nosso objetivo aqui será o de explorar algumas propriedades

lógicas elementares das proposições fitosociológicas imprecisas. Em particular, descreveremos a estrutura lógica das sentenças fitotaxonômicas elementares e observaremos algumas relações inferenciais nelas existente. Especial atenção será dada a generalização fitosociológica relacionando os requisitos de habitat de sintaxa, dado que estas expressões constituem-se em bases para a predição , a qual é uma das mais importantes ferramentas da ciência.

3.3.2 Sentenças fitosociológicas com predicados fuzzy

Sentenças atômicas

Proposições afirmativas de que a fitocoenose p (do nome **p**) possuem as características do tipo T podem sempre ser reparafraseadas da forma

$$p \text{ é } T$$

Como vimos (no capítulo 1), uma sentença desta forma é dita atômica ou elementar. Na semântica tradicional é assumido que esta proposição represente a afirmação de que o elemento p pertença ao conjunto dos elementos que possuam o predicado T. Se T é um predicado fuzzy, isto é, um rótulo linguístico de um conjunto fuzzy, então a interpretação da expressão acima possui uma forma diferente. Assumimos, depois de (Zadeh 1975a), que seu significado pode ser expresso através de uma equação de atribuição relacional:

$$R(tc(p)) = S(T)$$

onde $R(tc(p))$ é uma restrição fuzzy sobre a variável $tc(p)$. A proposição atômica fuzzy induz, em geral, uma distribuição de possibilidade $\Pi_{tc(p)}$ a qual postula-se ser igual a $R(tc(p))$ (Zadeh 1978):

$$\Pi_{tc(p)} = R(tc(p))$$

Existem algumas relações inferenciais entre proposições sinsistemáticas elementares, cuja evidência empírica é a disposição do fitosociologista em aceitar determinadas sentenças embasando-se em outras.

Considere as seguintes proposições q e r:

$$q = p \text{ é } S$$

$$r = p \text{ é aproximadamente } S$$

Da proposição q afirmando que a fitocoenose p representa o tipo S segue a proposição r que afirma que p lembra muito o tipo S. Em outras palavras: se p compartilha as caracteristicas que definem S (assim como afirma a proposição q), então p compartilha as características que definem outros tipos em menor grau de características representativas de S (assim como propõe r). Diagnósticos precisos de q contêm os diagnóticos de r em menor grau de acurácia. Analogamente, das proposições "p é um empobrecido fragmento de S" nós podemos inferir as proposições q e r como verdadeiras. As relações entre sentenças atômicas podem ser descritas de um modo formal utilizando o conceito de inclusão semântica introduzido por Zadeh (1978).

Princípio de mínima especificidade r é semanticamente contida por q (o que é denotado por $q \rightarrow r$) se e somente se

$$\Pi(q) \subset \Pi(r)$$

onde $\Pi(q)$ e $\Pi(r)$ são distribuições de possibilidades induzidas pelas proposições q e r respectivamente. O simbolo $\subset$ foi utilizado aqui no sentido de inclusão de conjuntos fuzzy. Pela definição do significado de "aproximadamente S" e "um empobrecido fragmento de S" teremos:

p é um empobrecido fragmento de $S \rightarrow p$ é S

p é $S \rightarrow p$ é aproximadamente S.

Condicionais

As sentenças discutidas anteriormente têm uma forma elementar - elas não contêm conectivos de nenhuma espécie. Condicionais sobre proposições complexas que contêm modificadores linguísticos (também chamadas sentenças moleculares) são de grande importância. Elas constituem a forma lógica de muitas generalizações fitosociológicas (Pickett e Kolasa 1989). Consideremos generalizações descrevendo requisitos de habitat para síntaxa. Elas são usualmente formuladas em linguagem natural e à primeira vista não lembram qualquer espécie de implicação :

- "Fitocoenoses desta associação [*Potentillo albae-Quecetum*] crescem em solos moderadamente férteis e relativamente secos [...]; o substrato é facilmente permeável e possui uma reação quase neutra, enquanto as camadas mais superficiais são fortemente ácidas em particular. (Matuszkiewicz).

– "Ela [*Trifido-Distichetum*] cresce [...] em habitats onde a cobertura de neve não persiste muito tempo e nas quais não forem excessivamente úmidas pelo fluxo das águas da chuva [...]. Ela requer, pelo menos nas camadas superfíciais do solo uma quantidade de carbonato de cálcio e uma reação fortemente ácida (pH 4-5)." (Pawlowski)

– "Em águas com correnteza moderada ou levemente corrente a associação *Potamo-Ranunculetum fluitantis* desenvolvem-se ....". (Pawlowski)

Nas citações acima, termos difusos (os quais foram sublinhados) descrevem de modo aproximado os fatores de habitat como mistura do solo e fertilidade, persistência da cobertura de neve etc. Estes termos podem ser tratados como sendo valores de variáveis linguísticas. Expressões como "ácido","fortemente ácido", "neutro", "em torno de 5", etc. podem então ser valores da variável linguística *acidez do solo.* Portanto, podemos tomar seus significados como sendo conjuntos fuzzy no universo $U = [0, 14]$.

Retornemos agora às generalizações que descrevem preferências de habitat de sintaxa. Assumiremos que cada sentença citada é de fato uma conjunção de condicionais implícitos. Observemos, por exemplo, a seguinte paráfrase da primeira das citações acima:

> Se p é *Potentillo albae-Quercetum* então p cresce em solos moderadamente férteis e, se p é *Potentillo albae-Quercetum* então p ocorre em solos *relativamente secos* e, ...

Cada condicional na conjunção acima assegura que se a variável linguística tipo de comunidade assumir o valor T, então a variável linguística que descreva os fatores de habitat tomará o valor W, cujo significado é $W \subseteq U$

$$\text{Se } p \text{ é } T \text{ então } u \text{ é } W \tag{3.18}$$

onde p é o nome de uma fitocoenose, u é o nome de um elemento de U, que é o universo das variáveis linguísticas que descrevem fatores ambientais. O termo T envolvido na fórmula (3.18) é um nome básico de S, desde que as características de habitat sejam obtidas apenas por "tipos puros" de sintaxa. Temos portanto

$$\text{Se } p \text{ é } S \text{ então } u \text{ é } W. \tag{3.19}$$

Por simplicidade, denotaremos expressões da forma (3.19) como

$$S \Rightarrow W. \tag{3.20}$$

A proposição (3.20) induz a distribuição de possibilidade (Zadeh 1978a, 1978b)

$$\Pi = \neg(S \times U) \oplus (A \times W), \tag{3.21}$$

onde A é a conjunto de todas as fitocoenoses descritas, S é o significado do nome S; $\neg$ é a negação (ou complementar), $\times$ e $\oplus$ denotam o produto

cartesiano e a soma limitada[2], respectivamente. A distribuição de possibilidade induzida por (3.21) é numericamente igual a uma relação fuzzy em $A \times U$.

3.3.3 Fito-Indicação : Um caso de predição

A maior parte das aplicações práticas da fitosociologia é baseada no fato de que toda associação de plantas corresponde à condições de habitat específicas. O procedimento da predição de fatores ambientais baseados na ocorrência de comunidades de plantas é chamado de *fito-indicação* . No prática fitosociológica usual as diferenças discutidas possuem natureza *entimemática*. Sua interpretação lógica será proposta a seguir.

O conhecimento referente às relações entre vegetação e cada fator de habitat pode ser expresso na forma de uma expressão condicional múltipla

$$(S_1 \Rightarrow W_1) \wedge \ldots \wedge (S_i \Rightarrow W_i) \wedge \ldots \wedge (S_n \Rightarrow W_n). \qquad (3.22)$$

onde $\wedge$ é o conectivo fuzzy E, S_i são os nomes de sintaxon S_i, e w_i os valores da variável linguística que descreve os parâmetros ambientais associados com a comunidade do sintaxon S_i; o significado de W_i é $W_i \subseteq U$; $i = 1, \ldots, n$.

A conjunção (3.22) é a base nomológica das predições em consideração . Suponha que a comunidade representativa do sintaxon S_i ocorre numa dada área. Se conectivos "*Se ... então ...*" e "*E*" forem interpretados de acordo com a Lógica Clássica bivalente, então a inferência pode ser efetuada da seguinte forma

[2]A definição destes conceitos forma vistos no capítulo anterior

$$\frac{(S_1 \Rightarrow W_1) \wedge \ldots \wedge (S_i \Rightarrow W_i) \wedge \ldots \wedge (S_n \Rightarrow W_n) \qquad S_j}{W_j} \tag{3.23}$$

A fórmula acima é uma sentença tautológica no cálculo proposicional, portanto este raciocínio é uma dedução . Todavia, inferências com a segunda premissa contendo um termo T em $\mathcal{T}_{tc}$ a qual é um nome composto ($T \notin \mathcal{S}$) não satisfaz ao esquema (3.23). Desde que o diagnóstico da comunidade for da forma

$$p \text{ é } T, \tag{3.24}$$

onde T é *algum* termo de $\mathcal{T}_{tc}$, será dada uma "interpretação fuzzy" a conjunção (3.22). Em virtude de (3.21) e (3.22) (Baldwin e Pilsworth 1979) nós obtemos a distribuição de possibilidade Π induzida pela conjunção de implicações para um dado fator de habitat:

$$\begin{aligned} \Pi &= R \\ &= \cup_{i=1,\ldots,n}[\neg(S_i \times U) \oplus (A \times W_i)], \end{aligned} \tag{3.25}$$

onde $R \subseteq A \times U$, e $W_i \subseteq U$ é o significado do termo W_i. Se o diagnóstico da comunidade possui a forma (3.25), então a restrição fuzzy sobre a variável que descreve os parâmetros de habitat é determinada da forma

$$\mathrm{W} = M(T) \circ R, \tag{3.26}$$

onde $\mathrm{W} \subseteq U$, $M(T) \subseteq A$, e $\circ$ denota a composição de relações fuzzy. A

fórmula (3.26) é referida como a regra composicional de inferência (Zadeh 1975). E' o equivalente fuzzy de (3.23).

Vamos ilustrar as considerações anteriores através de um exemplo prático. Seja a reação do solo descrita por uma variável linguística *acidez do solo (as)*, o o conjunto $\mathcal{T}_{as}$ da forma

$$\mathcal{T}_{as} = \left\{ \begin{array}{ccc} \text{"}\textit{fortemente}\text{ ácido"}, & \text{"ácido"}, & \text{"}\textit{fracamente}\text{ ácido"}, \\ \text{"}\textit{neutro}\text{"}, & \text{"}\textit{alcalino}\text{"}, & \text{"}\textit{algo}\text{ ácido"}, \\ \text{"}\textit{algo neutro}\text{"}, & \text{"}\textit{algo alcalino}\text{"}, & \text{"}\textit{indeterminado}\text{"}, \\ \text{"}\textit{irrestrito}\text{"} & & \end{array} \right\}$$

O universo de discurso da variável *as* pode ser denotado por U. O significado de um termo $W \in \mathcal{T}_{as}$ é um conjunto fuzzy W $\subseteq U$. É conveniente expressar as funções de pertinência de conjuntos fuzzy em U em termo das funções básicas para as quais os parâmetros podem ser ajustados de tal modo que estejam de acordo com a intuição usualmente relacionada com o significado de cada descrição vaga da reação do solo. Uma função padrão pode ser definida por (Blinder 1993c):

$$\mu_W(x) = \left\{ \begin{array}{ll} F(\frac{\alpha - x}{\gamma}) & \text{se } x < \alpha \\ 1 & \text{se } \alpha \leq x \leq \beta \\ F(\frac{x - \beta}{\gamma}) & \text{se } x > \beta \end{array} \right. \tag{3.27}$$

onde $W \subseteq U$ e α, β, γ são parâmetros que determinam a forma da função e $F(x)$ é dada por

$$F(x) = e^{-x^2}.$$

Os parâmetros α, β, γ serão tomados como

$$
\begin{aligned}
\text{fortemente ácido:} &\quad \alpha = \beta = 4 &&, \gamma = 2 \\
\text{ácido:} &\quad \alpha = \beta = 5 &&, \gamma = 2 \\
\text{fracamente ácido:} &\quad \alpha = \beta = 6 &&, \gamma = 2 \\
\text{neutro:} &\quad \alpha = \beta = 7 &&, \gamma = 2 \\
\text{alcalino:} &\quad \alpha = \beta = 7.5 &&, \gamma = 2 \\
\text{algo ácido:} &\quad \alpha = 2, \beta = 6 &&, \gamma = 3 \\
\text{algo neutro:} &\quad \alpha = 6.5, \beta = 7.5 &&, \gamma = 3 \\
\text{algo alcalino:} &\quad \alpha = 7.5, \beta = 10 &&, \gamma = 3 \\
\text{indeterminado:} &\quad \alpha \to \infty, \beta \to \infty &&, \gamma = 1 \\
\text{irrestrito:} &\quad \alpha = \beta = 0 &&, \gamma \to \infty.
\end{aligned}
$$

Tabela 3.1. Conjuntos Fuzzy correspondentes às Descrições Linguísticas de acidez do solo

rótulo	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
acido	0.11	0.21	0.37	0.57	0.78	0.94	1.00	0.94	0.78	0.57	0.37	0.21	0.11	0.05	0.02	0.01	0.00
fort.acido	0.37	0.57	0.78	0.94	1.00	0.94	0.78	0.57	0.37	0.21	0.11	0.05	0.02	0.01	0.00	0.00	0.00
frac.acido	0.02	0.05	0.11	0.21	0.37	0.57	0.78	0.94	1.00	0.94	0.78	0.57	0.37	0.21	0.11	0.05	0.02
algo acido	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.89	0.78	0.64	0.50	0.37	0.26	0.17
neutro	0.00	0.01	0.02	0.05	0.11	0.21	0.37	0.57	0.78	0.94	1.00	0.94	0.78	0.57	0.37	0.21	0.11
alcalino	0.00	0.00	0.01	0.02	0.05	0.11	0.21	0.37	0.57	0.78	0.94	1.00	0.94	0.78	0.57	0.37	0.21
algo alc.	0.03	0.06	0.11	0.17	0.26	0.37	0.50	0.64	0.78	0.89	0.97	1.00	1.00	1.00	1.00	1.00	1.00
algo neutro	0.11	0.17	0.26	0.37	0.50	0.64	0.78	0.89	0.97	1.00	1.00	1.00	0.97	0.89	0.78	0.64	0.50
indeterm.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
irrestrito	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Se $\alpha = \beta > 0$, então a função de pertinência $\mu_W(u)$ é da uma curva em forma de sino cujo valor central é α e cuja variação é γ. Se $\alpha < \beta$ então a função de pertinência possui um platô em $\alpha \leq u \leq \beta$. Note que "*indeterminado*" é o rótulo do conjunto vazio enquanto o significado do termo "*irrestrito*" é simplesmente U. É desnecessário frisar que, embora as funções de pertinência definidas por (3.27) e os parâmetros α, β, γ terem sido tomados de modo arbitrário, isto não significam que eles possam ser quaisquer outros. De fato, na teoria de conjuntos fuzzy, não há um modo particular de impor restrições à construção de funções de pertinência de um conjunto fuzzy, de modo que a escolha de tal função é algo subjetiva. Todavia, é claro que em toda aplicação os resultados dependerão do grau de acurácia da transição entre valores linguísticos para os conjuntos fuzzy. Alguns métodos de aquisição de funções de pertinência associados com

termos vagos foram propostos (por exemplo Labov 1973; Hersh e Caramazza 1976; Turksen 1988, 1991). Assim como não há uma regra simples e genérica a este problema, em qualquer aplicação séria das funções de pertinência às quais representem significados de alguns termos devem ser estimadas com extremo cuidado a fim de assegurar uma representação genuína.

Vamos assumir que U é o conjunto discreto de valores

$$U = \{2, 2.5, 3, \ldots, 9.5, 10\}.$$

Os conjuntos fuzzy correspondentes a cada termo em $\mathcal{T}_{as}$, calculados de acordo com (3.27), são apresentados na tabela 3.1.

Consideremos a conjunção (3.22) para três sintaxas para o exemplo considerado anteriormente neste trabalho, reparafraseado da forma

$$(S_1 \Rightarrow \text{algo alcalino}) \wedge (S_2 \Rightarrow \text{ácido}) \wedge (S_3 \Rightarrow \text{fortemente ácido}). \quad (3.28)$$

A relação fuzzy R, equivalente a distribuição de possibilidade Π, induzida pela proposição (3.4) é apresentada na tabela abaixo.

Tabela 3.2. Relação Fuzzy R equivalente à distribuição de possibilidade Π gerada pela proposição (3.12)

amostra	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
1	0.43	0.46	0.50	0.57	0.65	0.77	0.90	1.00	1.00	1.00	1.00	1.00	0.93	0.87	0.84	0.83	0.83
2	0.43	0.46	0.50	0.57	0.65	0.77	0.90	1.00	1.00	1.00	1.00	0.93	0.87	0.84	0.83	0.83	0.82	0.82
3	0.03	0.06	0.11	0.17	0.26	0.37	0.50	0.64	0.78	0.89	0.97	1.00	1.00	1.00	1.00	1.00	1.00
4	0.43	0.46	0.50	0.57	0.65	0.77	0.90	1.00	1.00	1.00	1.00	1.00	0.93	0.87	0.84	0.83	0.83
5	0.03	0.06	0.11	0.17	0.26	0.37	0.50	0.64	0.78	0.89	0.93	0.87	0.84	0.83	0.83	0.82	0.82
6	0.43	0.46	0.50	0.57	0.65	0.77	0.90	1.00	1.00	1.00	0.93	0.87	0.84	0.83	0.83	0.82	0.82
7	0.43	0.46	0.50	0.57	0.65	0.77	0.90	1.00	1.00	1.00	0.93	0.87	0.84	0.83	0.83	0.82	0.82
8	0.03	0.06	0.11	0.17	0.26	0.37	0.50	0.64	0.78	0.89	0.97	1.00	0.93	0.87	0.84	0.83	0.83
9	0.11	0.21	0.37	0.57	0.78	0.94	1.00	0.94	0.78	0.57	0.37	0.21	0.11	0.05	0.02	0.01	0.00
10	0.41	0.51	0.67	0.87	1.00	1.00	1.00	1.00	0.95	0.78	0.67	0.51	0.41	0.35	0.32	0.31	0.30
11	0.41	0.51	0.67	0.87	1.00	1.00	1.00	1.00	0.95	0.79	0.67	0.51	0.41	0.35	0.32	0.31	0.30
12	0.11	0.21	0.37	0.57	0.78	0.94	1.00	0.94	0.78	0.57	0.37	0.21	0.11	0.05	0.02	0.01	0.00
13	0.41	0.51	0.67	0.87	1.00	1.00	1.00	1.00	1.00	0.87	0.67	0.51	0.41	0.35	0.32	0.31	0.30
14	0.11	0.21	0.37	0.57	0.78	0.94	1.00	0.94	0.78	0.57	0.37	0.21	0.11	0.05	0.02	0.01	0.00
15	0.37	0.51	0.67	0.87	1.00	0.94	0.78	0.57	0.37	0.21	0.11	0.05	0.02	0.01	0.00	0.00	0.00
16	0.37	0.57	0.78	0.92	1.00	0.94	0.78	0.57	0.37	0.21	0.11	0.05	0.02	0.01	0.00	0.00	0.00
17	0.67	0.79	0.95	1.00	1.00	1.00	1.00	0.87	0.67	0.51	0.41	0.35	0.32	0.31	0.30	0.30	0.30
18	0.67	0.67	0.95	1.00	1.00	1.00	1.00	0.87	0.67	0.51	0.41	0.35	0.32	0.31	0.30	0.30	0.30
19	0.41	0.61	0.67	1.00	1.00	1.00	1.00	0.87	0.67	0.51	0.41	0.35	0.32	0.31	0.30	0.30	0.30
20	0.37	0.51	0.67	0.87	1.00	0.94	0.78	0.57	0.37	0.21	0.11	0.05	0.02	0.01	0.00	0.00	0.00

Predições da reação do solo obtidas por meio de regras composicionais de inferência embasando-se em vários diagnósticos de comunidades de plantas e na conjunção (3.4) são compilados na tabela seguinte. Os resultados aparentam ser bastante intuitivos.

Tabela 3.3. Predição da acidez do solo por meio da regra de inferência composicional $W = AL_{as}(S(T) \circ R)$

T - Premissa	W - Conclusão
S_1	algo alcalino
S_2	ácido
S_3	fortemente ácido
um fragmento empobrecido de S_2	ácido
um fragmento empobrecido de S_3	fortemente ácido
aproximadamente S_2	algo ácido
aproximadamente S_3	algo ácido
uma forma transacional entre S_1 e S_2	algo neutro
uma forma transacional entre S_1 e S_3	irrestrito
uma forma transacional entre S_2 e S_3	algo ácido
indeterminado	indeterminado

Devemos enfatizar, todavia, que o esquema de predição apresentado não clama para sí o direito de exclusividade. Não foi dito que todo o prognóstico de habitat possui a forma (3.26). Esta fórmula é apenas a entrada da mais simples variante de predição .

3.3.4 Discussão

Sentenças descrevendo relações entre vegetação e habitat são apesentadas neste trabalho sob a forma de condicionais não quantificados, embora eles sempre contenham algum quantificador fuzzy tal como: "comparativamente muito frequente", "usualmente", "algumas vezes". A linguagem PRUF (Zadeh 1978) permite a tradução deste tipo de sentenças; o racicínio aproximado baseado em premissas com quantificadores fuzzy foi discutido em (Zadeh 1983).

O Raciocínio Aproximado de acordo com a regra de inferência composicional pode ser entendido como uma "variação fuzzy" da forma simples de dedução nomológica (Hempel 1965).

Mais ainda, desde que o conhecimento na ciência da vegetação é impreciso por natureza, sistemas especialistas fuzzy podem servir de repositórios de conhecimeto fitosociológico, por serem capazes de armazenar premissas vagas e ou inexatas e de realizar inferências com as mesmas. A Lógica Fuzzy provê o substrato natural para o gerenciamento de tal conhecimento incerto. A componente representacional da Lógica Fuzzy, aonde as variáveis linguísticas são de vital importância, permitindo a tradução de conceitos inexatos e sentenças imprecisas através de conjuntos fuzzy (distribuições de possibilidade) sobre os quais as regras aproximadas de inferência podem ser aplicadas. A regra da inferência composicional e do princípio de inclusão em sistemas especialistas foi considerada por (Zadeh 1983). Todavia, sua formalização e aplicação prática pode ser encontrada em (Blinder e Bassanezi 1993) e a discussão detalhada de

suas possíveis aplicações em Ecologia Teórica podem ser encontradas em (Blinder 1993a,1993d; Blinder e Paes 1993). æ

3.4 Agruras da Inferência Fuzzy e outros azares

"Nem tudo é perfeito, ..."
(Anônimo, porém sábio.)

O principal e mais conhecido operador de implicação na Lógica Fuzzy é o da expressão (3.4):

$$I_m = A' \circ ((A \times B) \cup (\neg A \times B))$$

Também são bastante conhecidos e utilizados os operadores análogos aos da Lógica Polivalente, como os de Lukasievicz, Gogen, ...

Infelizmente, "nem tudo são flores". Foi argumentado no fim da seção anterior que o esquema de inferência (3.4) não é, *Graças a Deus*, o único esquema de inferência possível.

Ora, "*Infelizmente*" e "*Graças a Deus*" são expressões de significado *nítido*, qual o seu significado semântico de sua utilização aqui ? Explica-se:

Embora (3.4) funcione em muitos casos, desde 1979 conhece-se a partir de um artigo (Mizumoto et ali., 1979 *in:* Advances of Fuzzy Sets) que o operador (3.4) apresenta problemas em, pelo menos alguns casos de inferência.

Consideremos, por exemplo, um conjunto fuzzy A que é representado por um número fuzzy triangular no centro do intervalo discreto [1,10]. Uma representação para este conjunto fuzzy pode ser:

$$A = 0|1 + 0.25|2 + 0.5|3 + 0.75|4 + 1|5 + 0.75|6 + 0.5|7 + 0.25|8 + 0|9 + 0|10$$

e seja o conjunto fuzzy B o conjunto *verdadeiro* no mesmo domínio. B será representado por

$$B = 0.1|1 + 0.2|2 + 0.3|3 + 0.4|4 + 0.5|5 + 0.6|6 + 0.7|7 + 0.8|8 + 0.9|9 + 1.0|1$$

O significado semântico da implicação lógica $A \Rightarrow B$ pode ser "Se A for *aproximadamente* 5 então B é *verdadeiro*". Esta pode ser uma regra válida em algum contexto.

Porém, mesmo no caso em que A' for idêntico à A ($A' = A$) a inferência resultante é :

$$B' = 0|1 + 0.25|2 + 0.5|3 + 0.75|4 + 1|5 + 0.75|6 + 0.5|7 + 0.25|8 + 0|9 + 0|10$$

que não é coincidente com o consequente (B), de modo que este é um exemplo de que (3.4) não respeita, obrigatóriamente, *Modus Ponens* Clássico.

É possível mostrar que (3.4) não respeita também outras propriedades semânticas desejáveis para a implicação fuzzy.

Analisaremos em seguida que propriedades são semanticamente desejáveis para a Implicação Fuzzy, quais os principais operadores de implicação alternativos disponíveis na literatura e faremos uma análise de suas propriedades e impropriedades.

3.4.1 Propriedades semanticamente desejáveis para GMP

É razoável creditar impossível formular predicados nos quais um esquema de inferência fuzzy deveriam enquadrar-se pois se fosse possível, ele imporia uma rigidez ao esquema; uma vez que a inferência fuzzy deve ter suas raízes na subjetividade (embora as nutra de clara objetividade). Porém, podemos formular uma tabela de propriedades desejáveis para a inferência fuzzy, colocando as opções mais úteis para cada enunciado.

Uma tabela que contenha tais propriedades desejadas pode ser a seguinte:

Tabela de Propriedades desejáveis para o *Modus Ponens Generalizado*

Propriedade	Premissa	Dedução Desejada
GMP1	x é A	y é B
GMP2a	x é *muito* A	y é *muito* B
GMP2b	x é *muito* A	y é B
GMP3a	x é *mais-ou-menos* A	y é *mais-ou-menos* B
GMP3b	x é *mais-ou-menos* A	y é B
GMP4a	x é *não* A	y é *desconhecido*
GMP4b	x é *não* A	y é *não* B
GMP5	x é *não* B	y é *não* A

onde a inferência fuzzy representada pela tabela acima é derivada de um enunciado de implicação fuzzy do tipo "Se x é A então y é B", onde A e B são conceitos fuzzy, representados pelas suas respectivas distribuições de possibilidade.

Em geral, todas as propriedades de rótulo GMPa são intuitivas e desejáveis para proposições da teoria fuzzy clássica. Inclusive, GMP1 é a formulação de *Modus Ponens* Clássico e GMP5 a de *Modus Tollens*.

É claro que GMP5 só deve ser plausível no caso em que A e B sejam conjuntos do mesmo domínio, já caracterizando assim a primeira evidência de uma propriedade que não é possível em todos os casos (e às vezes,

inclusive, não desejável).

As opções existentes em GMP2, GMP3 e GMP4 não só dependem do tipo de inferência desejado (veremos exemplos concretos de casos onde escolhemos GMP2b como propriedade necessária, ao invés de GMP2a, para a dedução no capítulo seguinte), mas a caracterizam.

Claro está, que do ponto de vista pragmático, se uma das alternativas for satisfeita a outra não será, o que resulta portanto em pelo menos duas consequências: Nenhum esquema formal clássico satisfaz (ou poderá satisfazer) **total e genericamente** qualquer tipo de inferência fuzzy generalizada. A escolha das propriedades deve ser feita caso a caso. A outra consequência é que *devemos escolher* um sistema formal que nos satisfaça, e mais que isto, conhecer portanto as propriedades que ele satisfaz de antemão.

3.4.2 Outros operadores de implicação fuzzy

Consideremos agora os aspectos sintáticos da implicação fuzzy.

Os principais operadores de implicação fuzzy, juntamente com sua definição na forma relacional são:

$$R_m = (A \times B) \cup (\neg A \times Y) \tag{3.29}$$

$$\mu_{R_m}(x,y) = (\mu_A(x) \wedge \mu_B(y)) \vee (1 - \mu_A(x))$$

$$R_a = (\neg A \times Y) \oplus (X \times B) \tag{3.30}$$

$$\mu_{R_a}(x,y) = 1 \wedge (1 - \mu_A(x) + \mu_B(y))$$

$$R_c = A \times B \tag{3.31}$$

$$\mu_{R_c}(x,y) = \mu_A(x) \wedge \mu_B(y)$$

$$R_s = A \times Y \rightarrow_s X \times B \tag{3.32}$$

$$\mu_{R_s}(x,y) = \begin{cases} 1 & \text{se } \mu_A(x) \leq \mu_B(y), \\ 0 & \text{se } \mu_A(x) > \mu_B(y) \end{cases}$$

$$R_g = A \times Y \rightarrow_g X \times B \tag{3.33}$$

$$\mu_{R_g}(x,y) = \begin{cases} 1 & \text{se } \mu_A(x) \leq \mu_B(y), \\ \mu_B(y) & \text{se } \mu_A(x) > \mu_B(y) \end{cases}$$

$$R_b = (\neg A \times Y) \cup (X \times B) \tag{3.34}$$

$$\mu_{R_b}(x,y) = (1 - \mu_A(x)) \vee \mu_B(y)$$

$$R_\triangle = A \times Y \rightarrow_\triangle X \times B \tag{3.35}$$

$$\mu_{R_\triangle}(x,y) = \begin{cases} 1 & \text{se } \mu_A(x) \leq \mu_B(y), \\ \frac{\mu_B(y)}{\mu_A(x)} & \text{se } \mu_A(x) > \mu_B(y) \end{cases}$$

$$R_* = A \times Y \rightarrow_* X \times B \tag{3.36}$$

$$\mu_{R_*}(x,y) = 1 - \mu_A(x) + \mu_A(x).\mu_B(y)$$

$$R_\square = A \times Y \rightarrow_\square X \times B \tag{3.37}$$

$$\mu_{R_\square}(x,y) = \begin{cases} 1 & \text{se } \mu_A(x) < 1 \quad \text{ou } \mu_B(y) = 1, \\ 0 & \text{se } \mu_A(x) = 1 \quad \text{ou } \mu_B(y) < 1. \end{cases}$$

As leis de inferência composicional utilizadas para os operadores acima é a composição max – min.

A exemplo dos diversos sistemas e operadores de implicação material presentes em Lógica Polivalente, os operadores acima foram propostos visando a atingir alguma semântica particular. Todavia, podemos listar também uma tabela de propriedades desejáveis para o operador de implicação , provenientes da mesma literatura, resumindo de alguma forma as propriedades que os autores de tais operadores visavam alcançar.

Tabela 3.4.

Axiomas desejáveis para a relação de implicação I

I1	Se $A \leq A'$ então $I(A,B) \geq I(A',B)$	[Baldwin, 1978]
I2	Se $B \geq B'$ então $I(A,B) \geq I(A',B)$	[Baldwin, 1978]
I3	$I(0,B) = 1$	[Baldwin, 1978]
I4	$I(1,B) = B$	[Baldwin, 1978]
I5	$I(A,B) \geq B$	[Yager, 1980/177]
I6	$I(A,A) = 1$	[Bandler e Kohout, 1980/13]
I7	$I(A,I(B,C)) = I(B,I(A,C))$	[Trillas e Valverde, 1985/155]
I8	$I(A,B) = 1 \iff A \leq B$	[Gaines, 1976/77]
I9	$I(A,B) = I(c(B),c(A))$	[Bandler e Kohout, 1980/13]
I10	I é contínua	

Tabela 3.5.
Operadores de implicação mais frequentes na Lógica Fuzzy com inspiração nas Lógicas Polivalentes Clássicas

Operador proposto por	Forma	Propriedades satisfeitas
Zadeh	$\max(1-x, \min(x,y))$	I2,I3,I4,I10
Lukasiewicz	$\min(1, 1-x+y)$	I1-I10
Kleene-Dienes	$\max(1-x, y)$	I1-I5,I7,I9-I10

3.4.3 Análise das propriedades satisfeitas pelos operadores de implicação fuzzy usuais

Exibimos na tabela abaixo os resultados hoje conhecidos na literatura a respeito dos operadores de implicação acima, em termos das propriedades desejadas para *Modus Ponens Generalizado*. O símbolo "x" denota que a propriedade não é satisfeita e "v" denota sua satisfabilidade.

Esquema de inferência fuzzy	Antecedente	Consequente	R_m	R_a	R_c	R_s	R_g	R_b	$R_\triangle$	R_*	$R_\square$
GMP1	A	B	x	x	v	v	v	x	x	x	x
GMP2a	muito A	muito B	x	x	x	v	x	x	x	x	x
GMP2b	muito A	B	x	x	v	x	v	x	x	x	x
GMP3a	$\pm$ A	$\pm$ B	x	x	x	v	v	x	x	x	x
GMP3b	$\pm$ A	B	x	x	v	x	x	x	x	x	x
GMP4a	$\neg$ A	desconhecido	v	v	x	v	v	v	v	v	x
GMP4b	$\neg$ A	$\neg$ B	x	x	x	x	x	x	x	x	x

3.4.4 Discussão

Muitas críticas ao controle fuzzy têm sido recentemente feitas. Porém, como é que o Metrô japonês funciona a base de trens que utilizam Lógica Fuzzy em seus controladores se tais métodos são tão *não confiáveis* ? É porque foram utilizados os conectivos apropriados e escolhidos criteriosamente, a partir da análise de suas propriedades.

O que acontece é que isto torna difícil sua utilização em larga escala, ao não iniciado, por não haver algorítmo simples de escolha do conectivo apropriado. Muitos resultados e roteiros são disponíveis atualmente na literatura, mas um tanto inacessível a nossos propósitos, entre eles a popularização da Lógica Fuzzy a aplicações nas Ciências Biológicas.

Buscamos, na próxima seção , ao propor o uso da implementação da Lógica Fuzzy em Redes Neurais Artificiais, poder contribuir de alguma forma para simplificar este estado de coisas.æ

Capítulo 4

Lógica Fuzzy Neural

A finalidade deste capítulo é a de mostrar como podemos implementar a inferência fuzzy através de redes neurais artificiais e exibir exemplos de suas aplicações práticas a problemas biológicos.

Para tanto, exibimos inicialmente o modelo neural artificial enfocando sua inspiração biológica e depois concentrando-nos nas suas implicações formais.

Em seguida, discutimos as propriedades genéricas dos sistemas inteligentes, distinguimos suas formas de aprendizado, e exibimos os princípios formais abstratos do aprendizado através de redes neurais artificiais.

Nas seções seguintes, tratamos do aprendizado supervisionado em redes neurais artificiais, concentrando-se no algorítmo de Retro-Propagação do erro, exibindo uma demonstração para o mesmo e algumas simulações numéricas, algébricas e computacionais a título de ilustração .

Estabelecidas as bases do aprendizado supervisionado em redes neurais artificiais, passamos a tratar do assunto da implementação da inferência fuzzy através do mesmo. Exibimos também aqui exemplos de simulações computacionais a título de ilustração e verificação do funcionamento do modelo implementado.

A seguir mostramos como é possível implementar o sistema especialista para fito-indicação proposto na seção anterior através da arquitetura proposta. Exibimos os resultados obtidos através da simulação computacional dos dados das seções anteriores e comparamos seus resultados.

Finalmente, realizamos a conclusão do trabalho com base nos resultados obtidos e de suas perspectivas.

O objeto da Computação Neural são as Redes Neurais Artificiais. Estas são estruturas que mimetizam o processamento distribuido e massivamente paralelo, a exemplo do realizado em nosso cérebro. A conceituação exata das redes neurais é todavia uma tarefa difícil, posto sua implicação nas mais diversas àreas: Inteligência Artificial, Física, Computação , Biologia e inclusive na Filosofia e em Matemática Aplicada.

Podemos definir Redes Neurais Artificiais como sendo sistemas computacionais organizados a partir de elementos de processamento simples (neurônios artificiais) interconectados que processam informações através de uma dinâmica de adaptação aos estímulos externos, isto é, são sistemas capazes de capturar a dinâmica que associa um conjunto de entradas a um conjunto de saídas, sem que se conheçam as leis que os relacionem. Tais redes são modelos matemáticos, onde os elementos de processamento são unidades simples, com algumas características semelhantes aos neurônios biológicos.

Definiremos tais conceitos mais precisamente do ponto de vista matemático mais adiante.

4.3 Modelagem Neurofisiologica Simplificada

4.3.1 Inspiração da Neurofisiologia

Como o termo "Rede Neural" implica, seu apelo inicial foi dado pela tentativa de modelagem de redes de neurônios **reais** existentes no cérebro. Os modelos são extremamente simplificados quando vistos do ponto de vista neurofisiológico, embora ainda possamos acreditá-los como "metáforas" válidas dos princípios da computação "biológica". Da mesma forma que partes isoladas de um navio grande são irrelevantes no comportamento do navio (por exemplo, seus botes ou o compartimento de carga), muitos detalhes das células nervosas são irrelevantes para o entendimento do comportamento *coletivo* das redes de células.

Neurônios

O cérebro humano é originalmente composto por algo cerca de 10^{11} **neurônios** (células nervosas) de vários tipos distintos.

A figura 4.1 apresenta um desenho esquemático de um neurônio típico. Redes de fibras nervosas estreladas são chamadas de **dendritos** que são conexos ao **corpo da célula** ou **soma**, que é aonde está localizado o núcleo celular. Uma única e longa fibra que estende-se do corpo da célula é chamada de **axônio**, que é extremamente ramificada em sua extremidade. Cada uma destas ramificações termina em uma **junção sináptica**, ou **sinapse**, conectada a outros neurônios. Os terminais receptores destas junções a outras células podem encontrar-se nos dendritos ou ainda diretamente no corpo da célula. Um axônio típico realiza alguns milhares

de conexões com outros neurônios.

A transmissão do sinal de uma célula a outra através da sinapse envolve um complicado processo químico no qual substâncias transmissoras específicas (os **neurotransmissores**) são enviados de um lado e recebidos do outro lado da sinapse. Seu efeito pode acentuar ou diminuir o potencial elétrico existente no corpo da célula receptora. Quando seu potencial alcança um determinado limiar, um pulso ou **potencial de ativação** de intensidade e duração determinados é enviado através do axônio. Diremos então que a célula foi "ativada". O pulso então se propaga através das ramificações do axônio até as junções sinapticas com as outras células. Após sua ativação , a célula deve esperar um determinado período chamado **período refratário** para poder ser ativada novamente.

4.3.2 Estrutura "Lógica" do Neurônio

Focalizaremos nesta seção a estrutura *lógica* de um neurônio. Observemos as analogias entre o Neurônio e seu esquema lógico, tal como dado na figura 4.2

- Existe uma unidade de processamento, o círculo denotado por i, que representa o corpo da célula, o *soma.*
- Um número de linhas conectadas, *logicamente*, ao soma. São representadas pelas linhas com flexas na figura.
- Cada canal de entrada é resultante da combinação de um dendrito e uma sinapse, e haverão tantos canais de entradas quanto sinapses conectadas aos seus dendritos.

- Os canais de entrada são ativados pelos sinais recebidos das *"caixas lógicas"* aos quais estão conectados. A natureza lógica destas pequenas caixas (nossos axônios pré-sinápticos), e é expressa pelo fato que elas pode servir tanto para ativar um canal (carregar um pulso) ou não ativá-lo (atividade abaixo do limiar de disparo no neurônio pré-sinaptico).

Cada linha de entrada possui um parâmetro w_{ij} associado, onde o subscrito i especifica o neurônio (pós-sinaptico) que estamos descrevendo enquanto o subscrito j indexa os vários canais de entrada de sinais do neurônio. O valor numérico de w_{ij} é a *eficácia sinaptica* das sinapses que chegam ao neurônio i, as quais são determinantes ao potencial de atividade pós-sinaptica (PAPS) que será adicionado ao soma i se o canal j for ativado. Existe apenas uma saída lógica por unidade, que expressa a saída relevante do nosso neurônio - se ele emite um pulso de ativação ou não.

A operação na unidade pode ser resumida como segue:

- A todo momento, algumas das entradas lógicas são ativadas.
- O soma recebe uma entrada (PAPS) a qual é a soma linear das eficácias w_{ij} dos canais que estiverem ativados.
- A soma do PAPS é comparada ao valor limiar do neurônio i e o canal de saída é ativado se ela excede o limiar, caso contrário, nenhuma saída é emitida.

Este processo pode ser formalizado posteriormente associando a cada caixinha um variável e_j a qual pode tomar os valores 1 e 0, indicando

quando o canal associado à caixa estiver ativo ($e_j = 1$) ou inativo ($e_j = 0$). O PAPS no neurônio i pode ser expresso agora como sendo a soma de *todas* as eficiências sinápticas nos canais de entrada conectados ao nosso neurônio multiplicado pela variável e correspondente, o que assegura que apenas canais ativos contribuem. Em outras palavras, denotando o PAPS no neurônio i por a_i,podemos escrever:

$$a_i = \sum_{j=1}^{N} w_{ij} e_j \tag{4.1}$$

onde N é o numero de caixinhas, isto é, neurônios pré-sinapticos.

A operação desta "pequena máquina" pode ser representada através da função verdade lógica

$$S_j = f(a_i > \theta_i) \tag{4.2}$$

onde f é uma função que vale 1 se $a_i > \theta_i$ e 0 caso contrário. A variável S_j indica, em linguagem neurofisiológica, quando um pulso acontecerá no axônio de saída. Desde que a função verdade (4.2) é computada a partir das variáveis e_j, que por sua vez também assumem valores 0 e 1, cada uma das variáveis pode ser considerada como sendo a função verdade de alguma expressão lógica. Isto nos leva diretamente ao modelo de neurônio formal de Mc Culloch e Pitts.æ

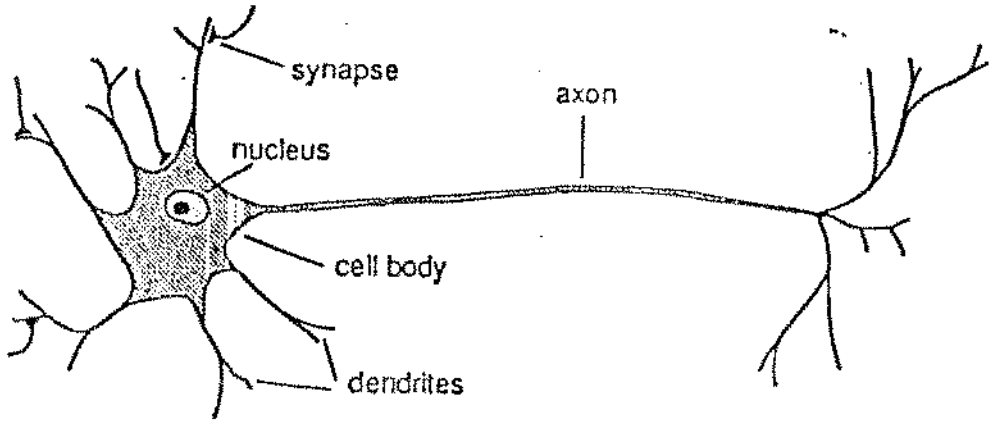


Figura 4.1: Representação esquemática de um neurônio

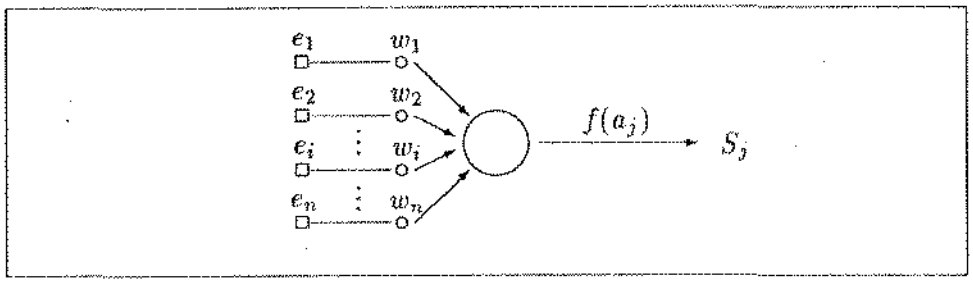


Figura 4.2:

A estrutura lógica de um neurônio. w_{ij} são as eficácias das sinapses que chegam ao neurônio j, representado pelo grande círculo. e_j são variáveis 0-1 representando a chegada ou não de algum pulso sobre o axônio pré-sináptico que conecta o neurônio j a outro, a_j é o chamado *potencial de ativação pós-sinaptico* e $f(a_j)$ é a função de ativação ou decisão do Neurônio. Se o neurônio "disparar" (ou não), S_j tomará o valor 1 (ou 0), respectivamente.

4.3.3 O Modelo Neural de Mc Culloch e Pitts

O Neurônio Formal Em 1943, Mc Culloch e Pitts num artigo bastante formal, mostraram que neurônios simplificados podiam ser combinados em pequenas sequências temporais, no final das quais haveria um simples neurônio na saída, cuja atividade seria o valor verdade de qualquer conectivo lógico binário, representados nos neurônios de entrada. Argumentavam então, que com tais redes elementares, poderiam ser construidas redes compostas as quais podiam realizar qualquer operação do Cálculo Proposicional.

Um elemento novo, essencial nesta construção , foi a introdução do *tempo*. Se os neurônios são "computadores" de expressões do Cálculo Proposicional, então deve-se levar em consideração o fato de que dos pulsos que cheguem aos axônios de dois neurônios que representem o valor verdade de duas proposições , então o neurônio que computa a operação lógica destas duas proposições na entrada pode dar sua resposta somente um *ciclo de tempo* posterior ao tempo de produção das duas entradas. Este ciclo de tempo, o qual será escolhido como unidade de tempo, é o tempo mínimo entre a emissão do pulso de entrada e da produção do pulso de saída (aproximadamente 1-2 milisegundos nas sinapses reais).

Basicamente, o *Neurônio Formal* de Mc Culloch e Pitts é a entidade descrita na seção anterior, usualmente com um número reduzido de canais de entrada. Cada neurônio possui seu próprio limiar e os canais possuem pesos numéricos. Os pesos e limiares são escolhidos de modo que a todo ciclo de tempo alguma operação lógica é realizada. Os arranjos mais

simples são exibidos de modo esquemático na figura 4.3. Eles realizam a computação da (a) negação de uma proposição (**não**), (b) da conjunção (**e**) de duas proposições e da (c) disjunção (**ou**).

Os diagramas podem ser lidos da seguinte maneira: em (a) o neurônio a esquerda emite um pulso no tempo t ($e(t) = 1$) se alguma proposição A é verdadeira. Se o canal possuir peso -1, como indicado, e o neurônio a direita possuir limiar $\theta = -0.5$, então no tempo $t + 1$ este neurônio não disparará. Se A for falsa, então o neurônio a esquerda não disparará no tempo t ($e(t) = 0$) e consequentemente o neurônio da direita disparará no tempo $t + 1$. Em outras palavras, o disparo do neurônio a direita representa a *negação* da proposição representada através do disparo do neurônio da esquerda. No diagrama (b) da figura 4.3, lê-se como: o neurônio da esquerda disparará em $t + 1$ se e somente se **ambos** os neurônios da esquerda dispararem no tempo t, indicando que as proposições afirmadas através dos neurônios da esquerda são **ambos** verdadeiros. Os parâmetros em (c) asseguram que o neurônio da direita disparará se pelo menos uma das suas entradas contiverem uma proposição verdadeira.

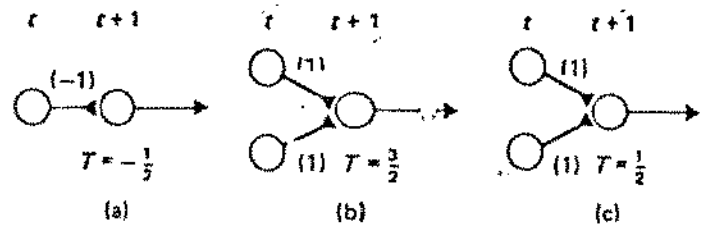


Figura 4.3: Três operações lógicas elementares: *(a) Negação (b) E (c) OU.* Em cada diagrama o estado do neurônio da esquerda é dado no tempo t e nos da direita em $t + 1$.

A elegância da Lógica Clássica reside no fato de que podemos construir qualquer operação booleana binária necessitando-se somente da negação e de apenas uma das duas operações binárias representadas na figura 4.3, e então, pela lei associativa, construir operções booleanas com qualquer número de proposições (como foi observado pela primeira vez no capítulo 1 do *Principia Mathematicae* de Russell). O único elemento adicional necessário é a função identidade, que provê um retardamento quando as saídas das operações prévias, que forem requeridas simultaneamente, sejam produzidas em tempos diferentes. Veja por exemplo, a figura 4.4. Desta forma, uma rede de neurônios formais que computem o *ou exclusivo* (**xor**) de duas proposições , as quais é verdadeira se **somente** um dos dois valores forem verdadeiros, pode ser como na figura 4.4. Note que se o valor verdade de duas proposições for representado a esquerda no tempo t, o resultado irá aparecer apenas no tempo $t + 3$. Note também que temos de recorrer ao uso da identidade para manter $A \wedge B$ enquanto $A \vee B$ estiver sendo negado.

É um simples exercício provar que nenhuma escolha de pesos e limiares possibilita que o neurônio formal compute o **XOR** em *um único* ciclo de tempo (Veremos uma demonstração deste fato na seção seguinte. A primeira demonstração publicada, e com detalhes , foi dada por Minsky e Papert, em 1969). O esquema da figura 4.4 é um exemplo de que é possível realizar tal computação em poucos passos. Isto decorre da suficiência dos conectivos exibidos na figura 4.3 para a construção de predicados arbitrários. É necessário, porém, ter em mente que a universalidade

formal requer tempo real, a qual aumenta a complexidade as proposições. Mais ainda, evidências neurológicas que necesitam-se de poucas dezenas de PAPS excitatórios para influenciar significativamente a probabilidade de que um pulso seja emitido pelo neurônio pós-sinaptico.

Faremos na seção seguinte uma outra abordagem, a geométrica, na qual poderemos ter mais claramente a importância do fato citado acima, possibilitando uma análise clara das possibilidades e restrições da neurônio formal de Mc Culloch e Pitts.æ

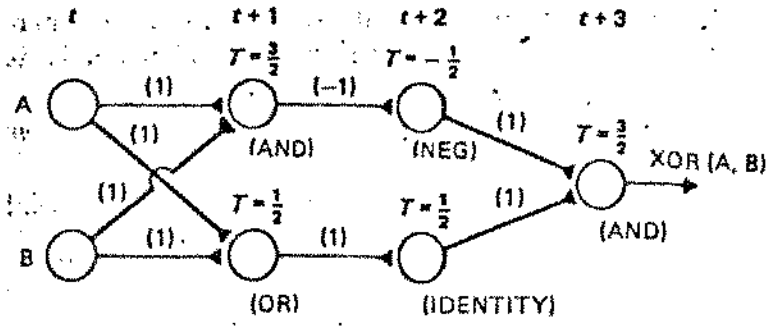


Figura 4.4: Uma rede construida para a computação do *OU Exclusivo* (**xor**) via $([A \wedge B] \vee [\neg[A \vee B]])$.

4.4 Análise Formal do Neurônio de Mc Culloch e Pitts

Neurônios de Mc Culloch e Pitts : Uma abordagem matemática

Neste item iniciaremos a abordagem mais matemática (menos biológica portanto) do modelo de Mc Culloch e Pitts , ressaltando quando possível sua capacidade lógica, seus vínculos com modelos históricos em Biomatemática e continuidade em nosso objetivo de apresentar as redes neurais do ponto de vista da Matemática abstrata.

O Neurônio de Mc Culloch e Pitts como um sistema tipo "Caixa Preta"

Pensando em termos de sistema, o modelo neuronal de McCulloch-Pitts (ou neurônio formal) pode ser imaginado como sendo um sistema tipo "caixa preta", caracterizado por apenas um número θ: o *limiar*. Esta "caixa preta" recebe "entradas" e responde com uma "saída". Típicamente, estas entradas são números em $[0,1]$ (monopolares) ou em $[-1,1]$ (bipolares) ou ainda em $\{0,1\}$ (discretas); cada uma das entradas tem um "peso" associado. Podemos pensar na "dinâmica" do neurônio formal como sendo uma função S que simplesmente responde ao critério: se a soma das entradas vezes os seus pesos for maior que o limiar, a resposta será 1 (ativação); de outra forma é 0 (desativação). A saída S do neurônio formal é portanto a função descontínua

$$S = \begin{cases} 0 \text{ se } \sum_{i=1}^{n} w_i e_i < \theta \\ 1 \text{ se } \sum_{i=1}^{n} w_i e_i \geq \theta \end{cases} \tag{4.3}$$

onde e_i são as entradas do neurônio formal, w_i os pesos respectivos, e θ é o limiar. Cabe mencionar, que a Teoria dos Dois Fatores de Nicholas Rashevski (um dos "pais" da Biomatemática), que é baseada numa certa contraposição de forças ying/yang – é muito similar ao neurônio formal.

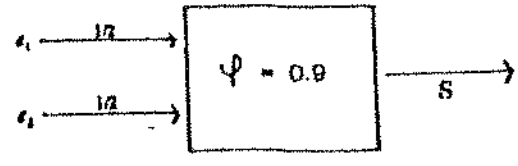
O Neurônio de Mc Culloch e Pitts como um Calculador Proposicional

Neste item forneceremos mais alguns exemplos de como o neurônio formal é capaz de realizar cálculos do valor verdade de sentenças proposicionais, mas usando a abordagem das tabelas verdade e do sistema *caixa preta*, como um primeiro passo para uma formalização mais abstrata.

Também aqui, as entradas e_i serão discretas: pertencerão á $\{0,1\}$. O zero será a representação do valor verdade **FALSO** (ou ainda o equivalente a desativado) e o um (1) o do **VERDADEIRO** (•, *ativado* respectivamente). Exibiremos a seguir algumas tabelas verdade do cálculo proposicional e o neurônio formal que realiza a operação equivalente:

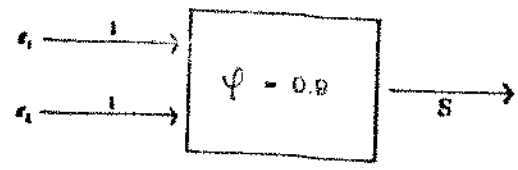
i) $e_1 \wedge e_2$:

e_1	e_2	$e_1 \wedge e_2$
0	0	0
1	0	0
0	1	0
1	1	1



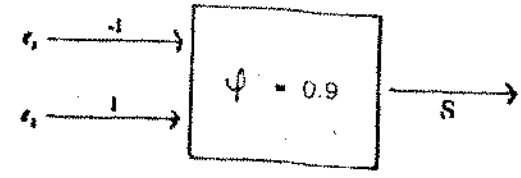
ii) $e_1 \vee e_2$:

e_1	e_2	$e_1 \vee e_2$
0	0	0
1	0	1
0	1	1
1	1	1



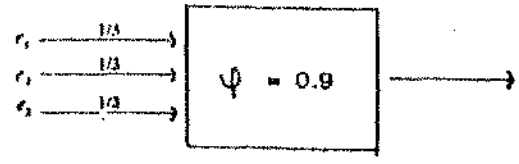
iii) $(\neg e_1) \wedge e_2$:

e_1	e_2	$(\neg e_1) \wedge e_2$
0	0	0
1	0	0
0	1	1
1	1	0



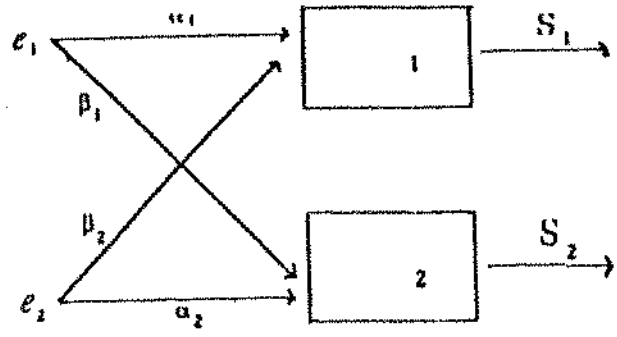
Os exemplos acima mostram como é possível implementar tabelas verdade de duas entradas (variáveis lógicas). Para tabelas de mais variáveis, o esquema geral é análogo; por exemplo:

$e_1 \wedge e_2 \wedge e_3$:



Como uma outra aplicação do neurônio formal, desenvolvida na mesma década que o trabalho de Mc Culloch e Pitts - ambos publicados nas primeiras edições do *Bulletin of Mathematical Biology*, criado e editado por N. Rashevshi – cabe mencionar a rede de discriminação psicofísica de Herbert Landahl. O problema é obter os pesos $\alpha_1, \alpha_2, \beta_1$ e β_2 e os

liniares θ_1 e θ_2 de modo que a rede:



reaja da seguinter forma:

$$e_1 < e_2 \quad \text{então} \quad s_1 = 0 \text{ e } s_2 = 1$$

$$e_1 > e_2 \quad \text{então} \quad s_1 = 1 \text{ e } s_2 = 0$$

Isto é, a rede é capaz de decidir qual das entradas é a maior. A solução é simples:

$$\theta_1 = \theta_2 = 0$$

$$\alpha_1 = \alpha_2 = 1$$

$$\beta_1 = \beta_2 = -1$$

Observe: Embora esta seção aparente alguma redundância com a seção de modelagem biológica, ela começa a apresentar uma das diferenças cruciais – ainda um pouco confundidas – dos modelos de redes neurais artificiais: o enfoque abstrato e computacional através de um modelo exclusivamente formal com vistas a resultados matemáticos, ao

invés da utilização de modelos de maior realismo biológico, que buscam auxiliar a esclarecer o funcionamento neural (como é o caso do sistema **Genesis** de J. Bower, Caltech).

Note também que este modelo de neurônio, para McCulloch e Pitts, seria uma evidência para se dizer que o cérebro (constituido de neurônios) é capaz, sem experiência prévia, de realizar cálculos lógicos. O aprendizado através da experiência é uma verdade inegável; o que ainda não foi resolvido após mais de 24 séculos é a polêmica de Platão e Aristóteles sobre o conhecimento inato, isto é: qual informação ou conhecimento já está "programado" no cérebro das pessoas ao nascerem? Warren McCulloch e Walter Pitts tinham a idéia de que o cálculo de proposições (Lógica de Boole, e consequentemente aritmética binária) é conhecimento inato, para provar esta tese desenharam um modelo bastante simplificado do neurônio, capaz de efetuar cálculos lógicos. Há várias objeções bastante sérias à proposição de Mc Culloch e Pitts; por exemplo, um neurônio isolado de uma minhoca, é funcionalmente (ao que se sabe sobre o funcionamento do neurônio) similar a um neurônio isolado do cérebro de Boole. Porém, não seria sensato dizer que a minhoca faz cálculos lógicos; apesar de, talvez, ter a capacidade "inata" de fazê-los.

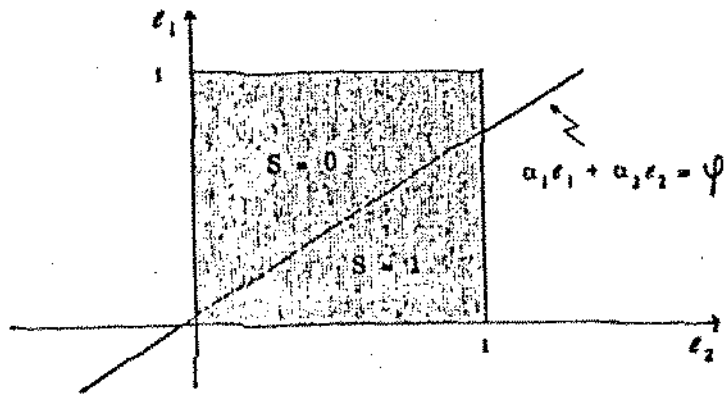
O modelo do neurônio, como desenhado por Mc Culloch e Pitts, mesmo que não "fosse a prova" do que queria-se demonstrar, transformou-se num dos paradigmas clássicos dos elementos básicos do processamento em paralelo das redes neurais.

4.5 Análise Geométrica do Neurônio de Mc Culloch e Pitts

Neste item faremos uma análise geométrica do neurônio de Mc Culloch e Pitts , de modo a permitir uma visualização de suas possibilidades e restrições . Por hora nos limitamos a exibir os pesos com os quais determinados resultados poderiam ser obtido, sem no entanto, discutir em quais condições um determinada família de conjuntos de pesos seria solução para o mesmo problema, ou ainda de que forma o "conhecimento" referente às tabelas verdade estão armazendas através deste conjunto e do limiar da unidade. Recorde-se que no caso biológico os pesos eram sinapses excitatórias ou inibitórias (pesos +1 e -1 respectivamente) préviamente fixadas, e na seção anterior foram exibidos pesos fracionários.

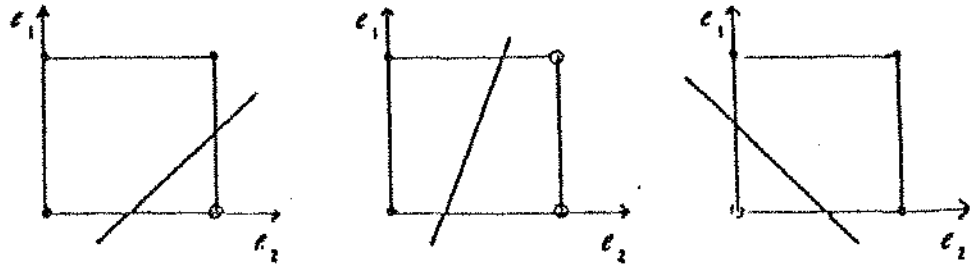
Para se entender e possibilitar uma análise geométrica do que faz (ou potencialmente pode fazer) o neurônio formal, vamos nos limitar a duas entradas monopolares, e_1, e_2 e $[0,1]$. Seja agora θ o limiar do neurônio formal em consideração . A reta $\alpha_1 e_1 + \alpha_2 e_2 = \theta$, com α_1 e α_2 fixos, divide o "plano de entradas" e_1, e_2 em duas regiões distintas: uma "acima" da reta e outra "abaixo" da reta. Numa delas vale $\alpha_1 e_1 + \alpha_2 e_2 > \theta$ e na outra $\alpha_1 e_1 + \alpha_2 e_2 < \theta$. Portanto o neurônio formal, no quadrado unitário $[0,1] \times [0,1]$, somente pode diferenciar *duas regiões separadas por uma*

reta: a reta $\alpha_1 e_1 + \alpha_2 e_2 = \theta$, como por exemplo na figura abaixo:



Devido a isto, diz-se que o neurônio formal somente pode diferenciar conjuntos *linearmente separáveis*.

Voltando a $e_1, e_2 \in \{0,1\}$, variáveis lógicas, e supondo que se queira separar os • dos o, o neurônio formal pode resolver todos os casos onde uma possível reta $\alpha_1 e_1 + \alpha_2 e_2 = \theta$ é ilustrada.



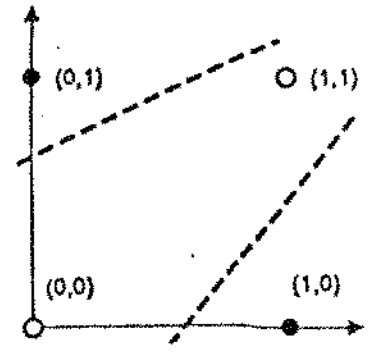
etc.

O caso do Ou-Exclusivo O Problema do *Ou-Exclusivo booleano*, ou da função **XOR**, consiste em computar um "sim", quando uma das duas entradas for verdadeira e um "não" caso contrário. Sua tabela verdade é a seguinte:

e_1	e_2	s_d
0	0	0
1	0	1
0	1	1
1	1	0

Tabela Verdade da função **XOR**

Fica claro a partir da figura acima que não é possível representar esta função atrvés de um único neurônio formal



uma vez que que os conjuntos os quais desejamos discriminar ($\{(0,0),(1,1)\}$ e $\{(0,1),(1,0)\}$) não são linearmente separáveis. Este é também o caso mais simples da *função paridade* de N entradas, discutido em detalhes no *Perceptron* (Minsky e Papert, 1969).

Todavia, vamos demonstrá-lo algébricamente. Desejamos que a função S descrita em (4.3) satisfaça simultaneamente as condições de contorno ($S(0,0) = S(1,1) = 0$ e $S(0,1) = S(1,0) = 1$). Omitindo-se o índice i, temos:

$$+w_1 + w_2 < w_0 \tag{4.4}$$

$$-w_1 - w_2 < w_0 \tag{4.5}$$

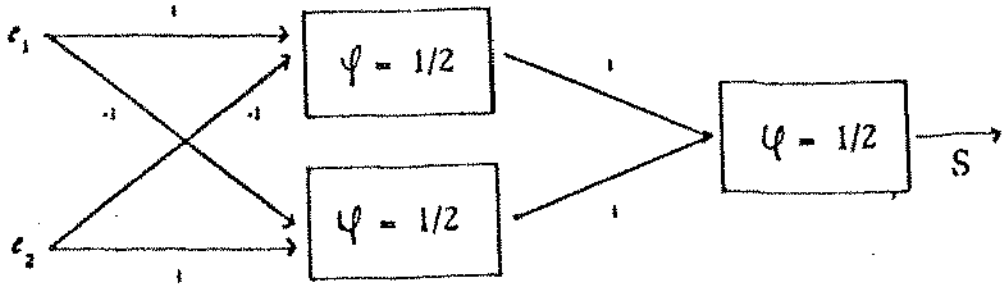
$$+w_1 - w_2 > w_0 \tag{4.6}$$

$$-w_1 + w_2 > w_0 \tag{4.7}$$

Combinando-se (4.4) e (4.7) temos que $w_1 < 0$, enquanto de (4.5) e (4.6) obtemos que $w_1 > 0$, as quais são impossíveis de ser satisfeitas simultaneamente. Pode-se mostrar de modo análogo que também é impossível satisfazer as restrições para os outros w_i's. Desta forma, fica demonstrado que uma unidade formal não pode realizar tal computação .

Também no problema de discriminação psicofísica há ainda uma situação similar a do ou-exclusivo; não é possível discriminar *três ou mais* entradas, somente com uma camada de neurônios formais. Estes dois problemas podem ser resolvidos com duas camadas de neurônios formais; as camadas de neurônios que estão entre os neurônios de entrada e os de saída são chamadas de *camadas escondidas* (ou *ocultas*). Este nome se deve ao fato de que estas camadas não recebem entradas do meio externo e nem provêem saídas diretamente a ele, sendo portanto, "ocultas" ao mesmo.

Rede para computar o XOR:



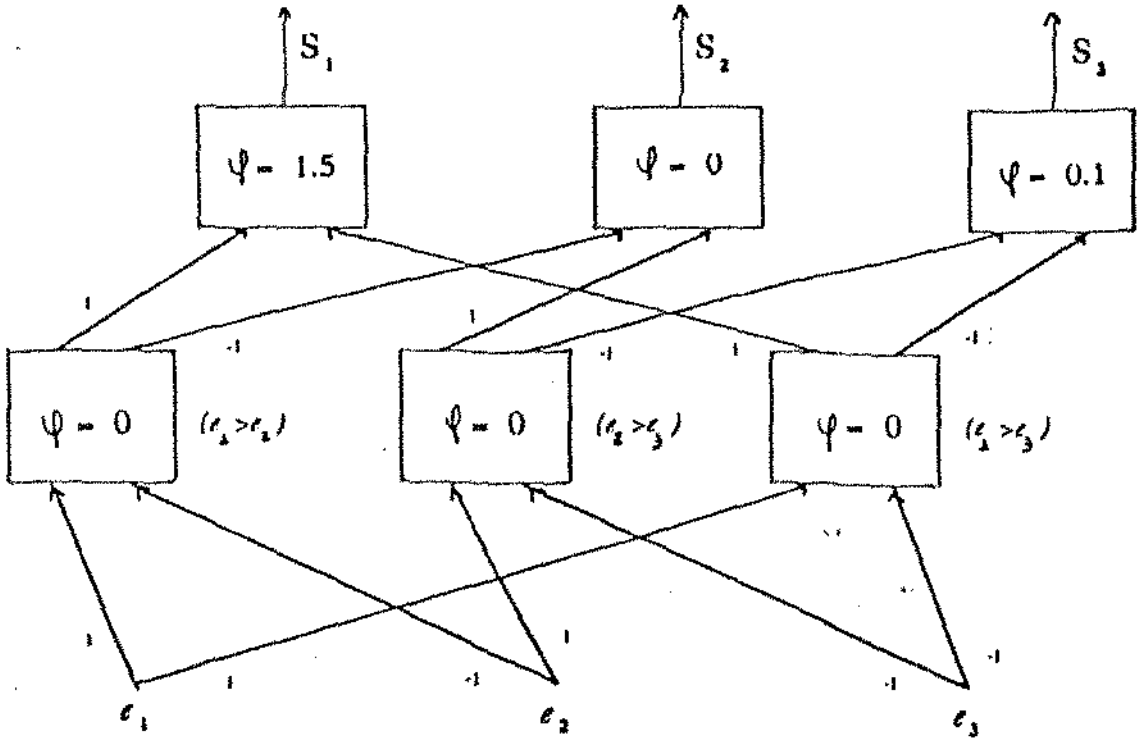
Discriminação de três entradas, *todas diferentes:*

Objetivo:

$$s_1 = 1, s_2 = 0, s_3 = 0 \text{ se } e_1 > e_2, e_1 > e_3$$

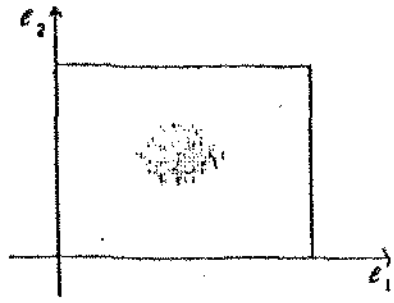
$$s_1 = 0, s_2 = 1, s_3 = 0 \text{ se } e_2 > e_1, e_2 > e_3$$

$$s_1 = 0, s_2 = 0, s_3 = 1 \text{ se } e_3 > e_1, e_3 > e_2$$

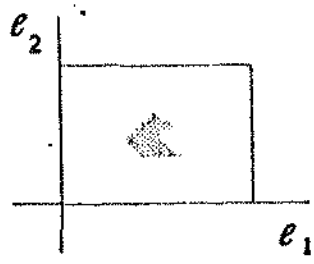


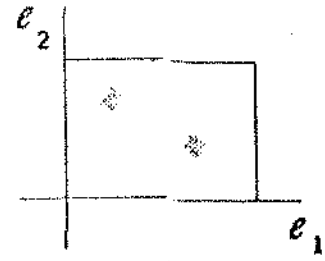
Redes neurais com, por exemplo, duas entradas que possuam *uma camada escondida*, são capazes de discriminar polígonos convexos no quadrado

unitário $[0,1] \times [0,1]$ do resto deste quadrado. Por exemplo:



A área hachurada acima pode ser discriminada do resto do quadrado unitário com uma rede neural que tem somente uma camada (entrada - saída), mas nas áreas hachuradas abaixo,





necessita-se de uma rede neural com mais de uma camada escondida para ser discriminada do resto do quadrado unitário.

4.6 Aprendizado em Redes Neurais

Podemos considerar como uma propriedade dos Sistemas Inteligentes sua capacidade de *aprender*, ou ainda, de *adaptar-se*. Dentre suas capacidades, podemos destacar as de fazer novas associações , formar novas categorias e novas regras além daquelas nele existentes, a partir da experiência.

Portanto, uma das consequências do aprendizado seria a *mudança de comportamento dinâmico do sistema*. Isto equivale dizer que o sistema teria a propriedade de auto-organizar-se, isto é, de *alterar seus parâmetros*. Tal mudança interna de parâmetros induz a uma variação quantitativa do comportamento do sistema e, eventualmente, qualitativa.

Outra boa e desejável propriedade de tal aprendizado seria "aprender bem". Aprender bem significa que o aprendizado deve refletir uma melhora da performance do sistema durante sua evolução dinâmica.

Das considerações efetuadas acima, podemos então formular duas proposições a respeito do aprendizado:

- **Proposição A1** Para haver *aprendizado* é necessário que haja alteração de seus parâmetros (forma na qual o sistema "codifica" ou representa o conhecimento). Como nas Redes Neurais Artificiais tal representação é efetuada através dos pesos das conexões sinápticas e de seus limiares, em geral representada por uma matriz de adjacência ou conexão M. Então o aprendizado será representado por:

$$\dot{M} \neq 0 \tag{4.8}$$

- **Proposição A2** O aprendizado será tanto melhor quanto mais próxima for a resposta dinâmica do sistema $o_{\text{saída}}$ de uma determinada saída esperada para o sistema o_{esperada}. Assim sendo, o aprendizado consiste em *minimizar* o erro $E(o_{\text{saída}}, o_{\text{esperada}})$.

Portanto, podemos concluir que para que haja aprendizado é preciso minimizar o erro na saída da rede e escolher o conjunto de pesos e limiares mais adequado. Mas como fazer isto?

Fornecemos alguns exemplos de pesos estabelecidos *a priori*, que podemos encontrar com um pouco de habilidade. Eles codificam a informação desejada por construção . Porém é facil notar que tais problemas constituem-se em excessão e não em regra e é extremamente desejavel encontrar um procedimento genérico.

4.6.1 Tipos de Aprendizado

Podemos sempre "ensinar" a rede a executar determinada tarefa através de procedimentos iterativos de ajuste das conexões w_{ij}. Isto pode ser feito basicamente de dois modos, dependendo das condições do problema que desejamos responder:

- **Aprendizado Supervisionado.** Aqui o aprendizado é feito com base na comparação direta entre as saídas da rede com as respostas corretas conhecidas. Algumas vezes também é chamado de *aprendizado com professor*. Nesta categoria também incluí-se o caso especial de *reforço do aprendizado*, onde o único feed-back dado é se a resposta está correta ou não, e não qual a resposta correta.

- **Aprendizado não supervisionado.** Algumas vezes aprendizado não é definido em termos de respostas a exemplos corretos específicos. A única informação disponível é a correlação entre os dados de entrada ou sinais. É esperado que a rede "crie" categorias a partir destas correlações , e produza sinais de saída correspondentes a categoria de entrada.

Existem sempre várias possibilidades de treinamento de uma rede para realizar alguma computação . Ao invés de se ter que especificar cada detalhe do cálculo, basta compilar um conjunto de exemplos representativos para treinamento. Isto significa que teremos esperanças de tratar problemas onde as regras são muito difíceis de serem conhecidas de antemão, como é o caso dos sistemas especialistas e da robótica. Isto também nos alivia de uma boa carga de tedioso (e caro) desenvolvimento de software, inclusive quando houverem regras explícitas. Como observou John Denker: "Redes Neurais são o segundo melhor jeito de fazer alguma coisa sobre qualquer coisa". Obviamente, a melhor maneira é encontrar e usar as regras corretas ("ideais" ou ótimas) ou um algorítmo de otimização para cada problema em particular, mas isto pode ser caro, dificultoso ou demorar muito. Portanto, a segunda melhor abordagem é baseada em aprendizado através de exemplos.

Quando o sistema possuir a capacidade de formar seus próprios padrões a partir dos estímulos de entrada, diremos então que ele possui *aprendizado não supervisionado.*

Se, no entanto, forem fornecidos ao sistema os padrões que desejamos que ele reproduza, então tal aprendizado será dito *supervisionado*. Para tanto, devemos fornecer ao sistema um conjunto de entradas e saídas desejadas para que ele "aprenda". Tal fato justifica a crença de que as Redes Neurais Artificiais são capazes de "aprender a partir de exemplos". Observe que para seu aprendizado fornecemos os padrões desejados e não as regras explícitas que associam o conjunto de entradas ao conjunto de saídas.

Este é um outro ponto importante, pois o aprendizado em RNA's é *não paramétrico*, isto é, não fornecemos ao sistema a forma da função que associa as entradas às saídas. As RNA's são capazes de aproximar qualquer função não linear de $\mathbb{R}^n \times \mathbb{R}^m$ (pelo Teorema de Kolmogorov), bastando para tanto ajustar convenientemente seus pesos e limiares. Apesar disto, não é possivel obter a forma explicita da função assim interpolada através de tais parâmetros. Todavia o Teorema de Kolmogorov é uma das garantias de que nossa busca para encontrar tais parâmetros não deve ser uma busca em vão, apesar de não nos fornecer como encontrá-los.

Vamos nos aprofundar um pouco mais nas duas proposições enunciadas no início desta seção e como elas se aplicam às redes neurais artificiais. Em seguida analisaremos a dinâmica do aprendizado das redes neurais e versaremos um pouco mais sobre o Teorema de Existência de Kolmogorov.

4.6.2 Plasticidade Neural (Proposição A1)

Conforme a proposição *A1*, para que haja aprendizado deve haver uma variação dos parâmetros do sistema. Porém, como se dá tal variação ? Qual é a regra a ser utilizada?

Pensando nestas questões, Donald Hebb, em 1949 no seu livro *"The Organization of the Behavior"* fez a seguinte proposta:

> "Quando um axônio de uma célula A estiver próximo o suficiente de excitar uma célula B ou repetidamente ou persistentemente tomar parte em ativá-la,um crescimento ou variação metabólica toma parte em ambas as células tal que a eficiência de A, enquanto uma das células que ativam B, é aumentada (p.62, tradução minha)".

A versão formal simplificada da regra de Hebb para a variação do peso da conexão w_{BA} entre o neurônio A, com uma taxa de ativação σ_A, projeta ao neurônio B, com uma taxa de ativação σ_B, é:

$$\Delta w_{BA} = \epsilon \sigma_A \sigma_B \tag{4.9}$$

onde ϵ é uma constante positiva. A *Regra de Hebb* é também conhecida na literatura como *Regra do Delta.*

A despeito da observação de Hebb ser fisiológica, a regra de variação de sinapse hebbiana foi precussora de vários algorítmos e inclusive de construção de dispositivos físicos. Ela pode ser encontrada hoje como o princípio básico dos algorítmos de condicionamento, das redes associativas [Kohonen, 1984] e das redes de correção de erro [Hopfield, 1984].æ

4.6.3 Função Erro (Proposição A2)

O termo **Função Erro** provêm especificamente da nomenclatura de ajuste de parâmetros (pesos e limiares) no sentido de mínimos quadra-

dos do erro. Porém, sua conceituação é análoga ao de muitas áreas da Matemática Aplicada. O nome mais geral da Função Erro, na teoria de sistemas dinâmicos, é *Função de Lyapunov*. Outros termos são o *Hamiltoniano*, encontrado no Cálculo Variacional e na Física-Matemática, *Função Energia* na Mecânica Quântica e na Mecânica Estatística. Na teoria de otimização encontraremos os análogos *Função Objetivo* ou *Função Custo* enquanto na Biologia Evolucionária usaremos o termo *Função de "Adaptabilidade"* (do inglês *fitness*).

É util pensar na analogia da função de erro como sendo uma superfície contínua e não linear sobre o espaço de parâmetros como a da figura 4.5. Tal superfície lembra uma região montanhosa.

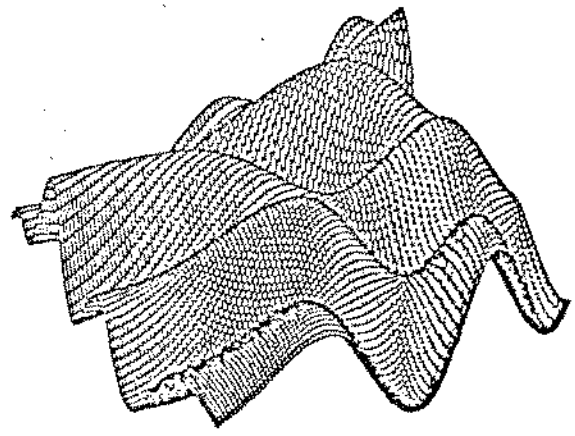


Figura 4.5: Representação metafórica da função erro como uma superfície não linear sobre o espaço de parâmetros.

De acordo com a proposição *A2*, para termos bom aprendizado é necessário minimizar o erro cometido pelo sistema para melhorar sua per-

formance. Isto equivale dizer que devemos "caminhar" a superfície de erro no sentido descendente até encontrar seu mínimo.

4.6.4 Dinâmica do Aprendizado

Podemos portanto imaginar a dinâmica do aprendizado como sendo a passagem do estado de nosso sistema para outro localizado mais abaixo na superfície da função erro.

Desta forma, os mínimos locais da função erro serão pontos atratores, que podem ser idealizados como os estados que melhor refletem as relações dadas pelo conjunto de treinamento (ou ainda como sendo os padrões armazenados pela rede, no caso das redes cuja dinâmica seja auto-associativa). Tal dinâmica pode ser comparada com a do movimento de uma bola sobre uma superfície montanhosa, sob a atuação da força da gravidade (que a puxa para baixo) e do atrito (que diminui sua energia). Iniciando de um ponto qualquer, a bola descerá montanha abaixo até encontrar algum dos mínimos locais (algum atrator). Os "vales" da superfície de erro (áreas de atração) em torno de cada mínimo correspondem às bacias de atração de um sistema dinâmico.

A figura 4.6 ilustra o funcionamento da dinâmica do aprendizado em Redes Neurais. A superfície de erro encontra-se desenhada sobre o espaço de todos os possíveis estados da rede (o **espaço de configuração**). Os mínimos locais da superfície de erro ζ_i^μ funcionam como *pontos atratores* neste espaço. A dinâmica do sistema leva estados iniciais para algum dos atratores, como exibido nas trajetórias desenhadas. Assim, todo o espaço de configuração é dividido em *bacias de atratores* dos diferentes

atratores (mínimos locais da superfície de erro). O desenho apresentado é, todavia, uma metáfora. Em particular, o espaço pode ser realmente apenas um conjunto discretos de pontos (em um hipercubo) e finito (ainda que grande), como no caso dos neurônios com analogia fisiológica da seção 3.1, e não regiões contínuas.

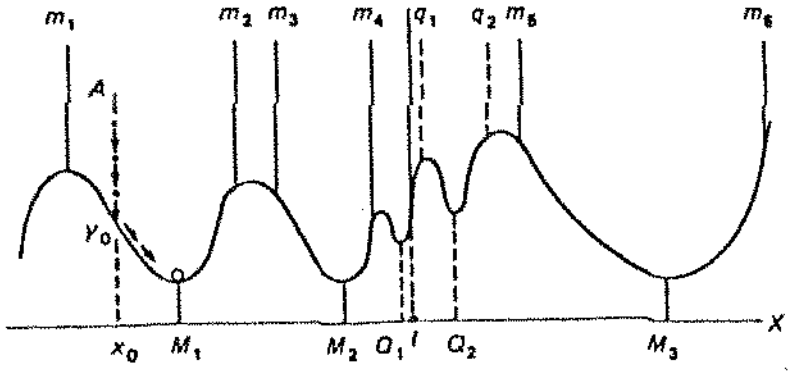


Figura 4.6: Representação metafórica em uma dimensão da dinâmica de aprendizado. Os pontos $M_1 - M_3$ são mínimos locais (pontos atratores), $Q_1 - Q_2$ são estados espurios e $m_1, \ldots, m_6$ são máximos locais que delimitam as bacias de atração .

Este exemplo constitui-se numa das possíveis analogias entre o aprendizado em redes neurais artificiais e a teoria de sistemas dinâmicos. Poderemos observar que de fato o aprendizado em redes neurais é uma aplicação direta do Cálculo Diferencial, na sua forma mais clássica de sistema dinâmico.

Um exemplo de aplicação : Memória Associativa

A dinâmica de aprendizado nos remete a um dos problemas centrais em redes neurais, que é o de como a rede pode armazenar informações , de que forma ela faz isto, e como podemos recuperá-los e eventualmente

generalizá-los.

A descrição formal do problema da memória associativa pode ser a seguinte:

> Armazenar um conjunto de p padrões η_i^μ de modo que quando apresentarmos um novo padrão ζ_i, a rede responda reproduzindo o padrão armazenado mais semelhante a ζ_i.

Ou seja, se fornecermos a rede a configuração próxima de ζ_i, desejamos saber qual (se houver) um conjunto de w_{ij}'s que faça a rede fornecer-nos o estado ζ_i. Uma ilustração pictórica disto seria a seguinte: suponhamos que codificamos em uma rede informações sobre grandes físicos e suas equações famosas. A entrada "$F = ma$" na rede deverá ser suficiente para recuperar informações a respeito de Newton, enquanto "$E = mc^{3n}$" deverá associar-se a Einstein, a despeito do erro de exatidão no padrão de entrada.

Aplicações comuns deste tipo de fato são a reconstrução e reconhecimento de imagens e a recuperação de informações bibliográficas a partir de referências parciais (como por exemplo o título incompleto de um artigo buscado).

4.6.5 Existência de Aproximação de Funções de $\mathbb{R}^n \times \mathbb{R}^m$ através de Redes Neurais Artificiais

O Teorema de Kolmogorov

Em 1957 o matemático Andrei Kolmogorov publicou um teorema relativo à representação de funções contínuas arbitrárias de $\mathbb{R}^n \to \mathbb{R}$ em

termos de funções de apenas uma variável [148]. Este teorema intrigou um grande número de matemáticos e nos vinte anos que se seguiram vários aperfeiçoamentos deste resultado foram efetuados, notavelmente os de G.G. Lorentz [163,213].

O Teorema de Kolmogorov foi descoberto durante um "amigável duelo" entre ele e V.I. Arnol'd no qual cada um deles tentava ser o primeiro a resolver as questões restantes acerca do 13.o problema de Hilbert (Matemático notável, que anunciou em 1900 uma lista de problemas a serem resolvidos pelos matemáticos do século 20. Alguns deles, ainda não foram resolvidos ...). Numa série de papers na metade da década de 50 eles disputaram sua batalha, resultando na "vitória" de Kolmogorov.

Embora o teorema de Kolmogorov seja extremamente poderoso no tocante a sua generalidade (é difícil acreditar a primeira vista que possa tal resultado!), ele não foi de grande valia para que outros matemáticos pudessem utilizá-lo no demonstração de outros teoremas importantes. Todavia, causou grande impacto na área de redes neurais.

O Teorema de Kolmogorov pode ser enunciado como segue:

> *Teorema 1* **Teorema de Existência de Aproximação de Funções através de Redes Neurais** *Dada qualquer função contínua* $f : [0,1]^n \rightarrow \mathbb{R}^m, f(\vec{x}) = \vec{y}, f$ *pode ser implementada exatamente através de uma rede neural de três camadas com* n *unidades de entrada na primeira camada* $(\vec{x})$, $(2n+1)$ *elementos de processamento na camada intermediária, e* m *unidades de processamento na camada de saída* $(\vec{y})$.

A prova deste resultado pode ser encontrada em [109], a qual não será apresentada aqui, por não apresentar nenhum insight técnológico útil, como será discutido.

Como enunciado no teorema, a *Rede de Kolmogorov* consiste em três camadas de elementos de processamento, onde as funções de transferência entre a camada de entrada e a intermediária são da forma

$$z_k = \sum_{j=1}^{n} \lambda^k \psi(x_j + k\epsilon) + k$$

onde λ é uma constante real e ψ é uma função real, contínua, monotonamente crescente e independente de f (embora elas dependam de n). A constante ϵ é um número racional $0 < \epsilon \leq \sigma$ onde σ é uma constante positiva arbitrária.

Os m elementos da camada de saída possuem a função de transição

$$y_i = \sum_{k=1}^{2n+1} g_i(z_k)$$

onde as funções g_i, $i = 1, 2, \ldots, m$ são reais e contínuas (e dependentes de f e ϵ).

Isto é tudo o que sabemos. Nenhum exemplo específico de função ψ nem de constante ϵ são conhecidos, nem alguma função g. A prova do teorema não é construtiva, de modo que ela não nos diz como obter estas quantidades. É estritamente um teorema de existência. Ela nos diz que uma rede com a topologia dada deve existir, mas não nos diz como encontrá-la. Bem, desde 1958 este resultado é amplamente conhecido, mas até agora nenhum método para encontrar tais quantidades foi proposto, e, ao que

parece, não o teremos tão cedo. Assim, o valor deste resultado é uma segurança intelectual de que funções vetoriais contínuas no cubo unitário (na verdade, este teorema pode ser extendido para qualquer *compacto*, isto é, todo conjunto fechado e limitado de dimensão finita) pode ser implementada utilizando-se uma rede neural de três camadas.

Implicações do Resultado

O Teorema de Existência de Aproximações de Funções por Redes Neurais de Kolmogorov é uma garantia de que nossa busca de aproximações de funções por meio de redes neurais, pelo menos em teoria, procede. Todavia, a utilização direta de seu resultado é duvidosa, posto que não se conhecem métodos construtivos para encontrar as funções g_i, necessária para a possível aplicação prática do teorema. æ

4.6.6 Considerações particulares ao trabalho

É facil ver que, apenas a partir das considerações anteriores, é possível elaborar uma quantidade enorme de modelos e embricamentos interessantes. Isto faz das redes neurais um campo bastante atrativo.

Porém, ao nosso modesto trabalho, será suficiente e necessário estudar o aprendizado supervisionado em redes de retro-propagação . As redes de retro-propagação são a versão melhorada das redes de unidades Mc Culloch e Pitts para possibilitar algorítmo de treinamento genérico. Tal estudo será levado a cabo a partir da seção seguinte.æ

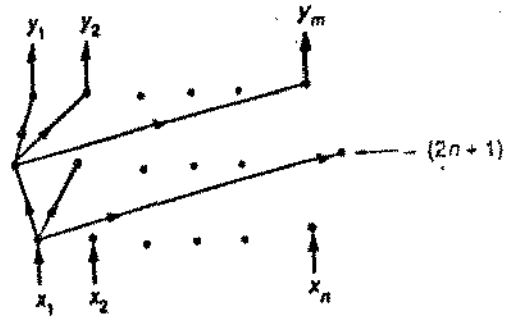


Figura 4.7: Topologia da rede de Kolmogorov. Esta rede pode implementar *exatamente* qualquer função contínua. Ela possí três camadas: A primeira possui n neurônios; a segunda possui $(2n+1)$ neurônios semi-lineares (suas funções de transferência são somas ponderadas lineares); a terceira possui m neurônios com funções de transferência não lineares.

4.7 Aprendizado Supervisionado em Neurônios Formais

As limitações do neurônio formal não se aplicam a redes de unidades com camadas intermediárias ou "ocultas" entre as camadas de entrada e saída. De fato, como será visto posteriormente, uma rede com apenas uma camada oculta pode representar qualquer função booleana (incluido o caso do **XOR**). Embora a grande potencialidade das redes neurais com múltiplas camadas sejá conhecida há tempo, apenas agora foram exibidos algorítmos que façam com que elas "aprendam" uma função particular, como o caso de "retro-propagação" e outros. A falta de tal algorítmo em conjunto com as demonstrações de Minsky e Papert *Perceptrons*, 1969) de que apenas funções linearmente separáveis pudessem ser representadas por uma único neurônio formal, levou a um arrefecimento do interesse em redes neurais até recentemente.

Esta seção tem por objetivo exibir, comentar e demonstrar o algorítmo de *retro-propagação* , que será utilizado ao longo deste trabalho como regra de aprendizado nas redes neurais aqui utilizadas.

4.7.1 Redes Neurais Artificiais como funções não lineares

Já com uma camada escondida e várias entradas, descobrir heuristicamente os valores dos pesos e limiares para que uma rede neural efetue uma dada tarefa pode ser bastante complexo. Devido a isto é importante ter-se um *algorítmo* para o cálculo destes parâmetros.

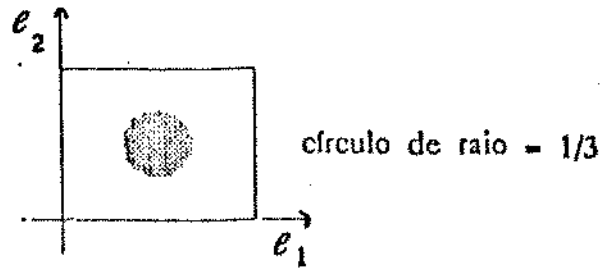
Formularemos, inicialmente, o problema genérico das redes neurais arti-

ficiais, que pode ser enunciado da forma seguinte:

> Seja Q^n o hipercubo $[0,1]^n$, supondo que se tem n entradas monopolares; seja $S^m = \{0,1\}^m$ o universo das possíveis "saídas". Supondo que se deseja ter m saídas, a "tarefa" que se propõe é encontrar uma função de Q^n em $S^m, F : Q^n \rightarrow S^m$; cujo objetivo é construir uma rede neural com n entradas e m saídas, de modo que ao se dar a entrada $(e_1, \ldots, e_n)$ a resposta da rede seja $F(e_1, \ldots, e_n)$ para todo $(e_1, \ldots, e_n) \in Q^n$.

Considere, por exemplo, um problema que possua duas entradas e uma saída, como o caso de discriminar se um ponto de $[0,1] \times [0,1]$ faz parte da circunferência de centro $(\frac{1}{2}, \frac{1}{2})$ e raio $\frac{1}{3}$. Tal função F poderia ser definida como

$$F(e_1, e_2) = \begin{cases} 1 & \text{se} \quad (e_1 - \frac{1}{2})^2 + (e_2 - \frac{1}{2})^2 < \frac{1}{9} \\ 0 & \text{se} \quad (e_1 - \frac{1}{2})^2 + (e_2 - \frac{1}{2})^2 > \frac{1}{9} \end{cases}$$



Observe que embora a definição da função F tal como acima descreva perfeitamente o problema, nenhuma regra para sua resolução dada. Este é o caso geral dos problemas que utilizam redes neurais.

Topologia da rede

O primeiro passo para a resolução do problema é decidir quantas camadas ocultas serão necessárias para resolver a tarefa e quantos elementos cada camada deverá conter. Isto é o que se conhece como sendo a *TOPOLOGIA* da rede, e em geral não há regras para resolver este problema; o que se pode fazer então é experimentar diferentes topologias até se obter a mais adequada. Mas, mesmo supondo que de alguma forma se descobriu qual seja a topologia adequada, ainda resta o problema de se obter os pesos e limiares que realizem a tarefa.

Existência A discussão acima somente tem razão de ser se for possível garantir que, para cada função $F : Q^n \rightarrow S^m$ de tarefas, exista uma rede neural com uma topologia e um conjunto de pesos e limiares que realizem a tarefa F). De fato, para o caso que será considerado (função de ativação sigmoide), pode-se provar que como consequência do *Teorema de Kolmogorov* tal rede neural positivamente existe – no entanto, como se podia esperar do "pai da Estatística moderna" – infelizmente a prova é de existência, e não nos fornece nenhuma indicação de como se possa construir tal rede.

Método de Resolução Para se descobrir a topologia da rede, como já foi dito, não se conhece algoritmo algum; assim, vamos supor que a topologia é conhecida, e que queremos obter o valor dos pesos e limiares. Para isto, deve-se notar que para cada conjunto de pesos $\vec{p}$ (um vetor) e para cada conjunto de limiares $\vec{\Theta}$ (um outro vetor), a rede é uma função

$N(\vec{p},\vec{\Theta}) : Q^n \rightarrow S^m$.

O que se deseja é obter $\vec{p}$ e $\vec{\Theta}$ de forma tal que para todo $\vec{e} \in Q^n : F(\vec{e}) = N(\vec{p},\vec{\Theta})(\vec{e})$. Isto sugere um problema de otimização:

Minimizar um erro do tipo $\sum_{\vec{e}\in Q^n} |F(\vec{e}) - N_{(\vec{p},\vec{\Theta})}(\vec{e})|$ em função de $\vec{p}$ e $\vec{\Theta}$.

Por um lado, a soma sobre $\vec{e} \in Q^n$ é infinita, por outro, as funções de saída são funções de Heaviside, portanto discontínuas. Assim sendo, a minimização por mínimos quadrados não é possível.

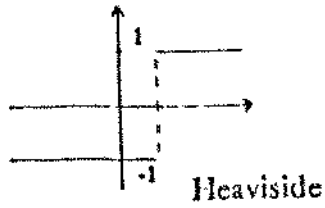
4.7.2 Conjunto de Treinamento

Para usarmos o método de mínimos quadrados é necessário que *a soma seja finita*, e que as *funções de saída sejam diferenciáveis*. Forçar uma soma finita quer dizer que o conjunto infinito Q^n deve ser substituido por uma amostragem finita de Q^n, isto é, um sub-conjunto finito de Q^n: o conjunto de treinamento da rede. Esta amostragem de Q^n ou conjunto de treinamento deve ser cuidadosamente escolhido de forma que "represente o mais fiel possível a função F". Desta escolha depende a capacidade da rede para "generalizar" o seu aprendizado a todo o Q^n; já que se deseja que a rede "treinada" realize a função F sobre todo Q^n com o menor erro possível.

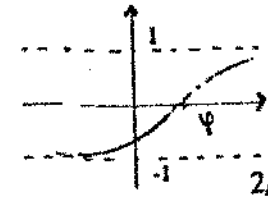
Funções de ativação Contínuas

Para substituir as funções de Heaviside por funções diferenciáveis, é útil trabalhar com entradas e saídas bipolares, com -1 sendo "apagado" e $+1$ sendo "aceso". Q^n então será $[-1,1]^n$, e $s^m = \{-1,1\}^m$. A escolha

de funções diferenciáveis "semelhantes" a função de Heaviside é variada. Por exemplo a Arco-Tangente:

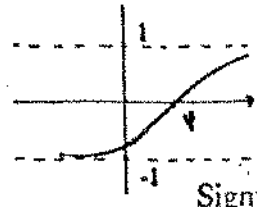


Heaviside

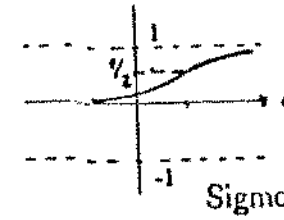


$2/\pi \operatorname{Arctg}(t-\psi)$

ou a sigmoide bipolar $S = \left(\frac{2}{1+e^{(-t+\theta)}}\right) - 1$ ou a sigmoide monopolar $S = \frac{1}{1+e^{(-t+\theta)}}$



Sigmoide bipolar



Sigmoide monopolar

É conveniente usar $s(t)$, já que a sua derivada é $\frac{ds(t)}{dt} = 1 - s^2$ se bipolar, $\frac{ds(t)}{dt} = s(1-s)$ se monopolar. Isto simplifica bastante os cálculos computacionais como veremos adiante.

Método de Treinamento: Minimização da Função Erro

Com estas simplificações e mudanças, o problema se reduz a minimizar a função erro E (ou custo):

$$E(\vec{p}, \vec{\Theta}) = \sum_{\vec{e} \in T} [F(\vec{e}) - N_{(\vec{p},\vec{\Theta})}(\vec{e})]^2$$

com $T \subset Q^n$, T finito. Note que o valor absoluto foi substituido pelo

quadrado; assim $E(\vec{p}, \vec{\Theta})$ é diferenciável com respeito a cada componente de $\vec{p}$ e de $\vec{\Theta}$.

Existem diversas técnicas de otimização não linear que minimizam $E(\vec{p}, \vec{\Theta})$. Uma das mais usadas em treinamento de redes neurais é a do decaimento pelo gradiente (Método do Gradiente Descendente); de fato, o método dos Gradientes Descendentes Conjugados é o que dá melhor resultado. Porém, o objetivo é tentar fornecer uma apresentação compreensível do método de decaimento pelo gradiente para treinamento de redes neurais; este método se conhece como propagação reversa (back propagation), ou simplesmente *retro-propagação* .

Simplifiquemos inicialmente a notação, escrevendo $(\vec{p}, \vec{\Theta}) = \vec{w}$, isto é, as primeiras componentes do vetor $\vec{w}$ correspondem a $\vec{p}$, e as últimas a $\vec{\Theta}$. O que se deseja é obter um mínimo para $E(\vec{w})$: Começamos com um vetor qualquer $\vec{w}_0$, e vamos "melhorar" a estimativa $\vec{w}_n$ da seguinte forma:

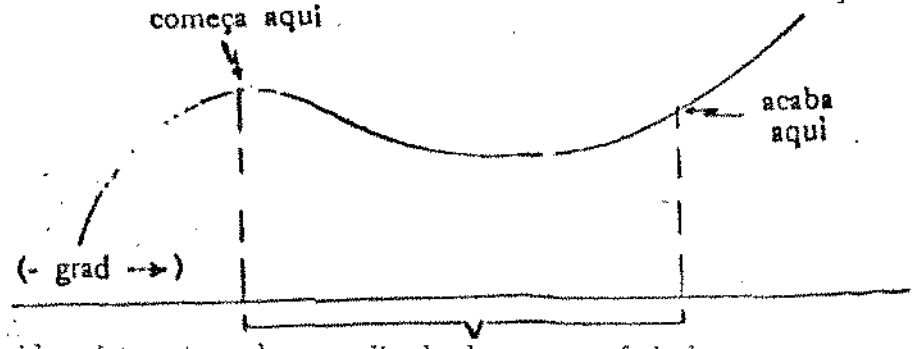
$$\vec{w}^{(t+1)} = \vec{w}^{(t)} - k \ \text{grad} \ E(\vec{w}^{(t)})$$

com k um número entre 0 e 1 (em geral, usa-se $k = 0,1$).

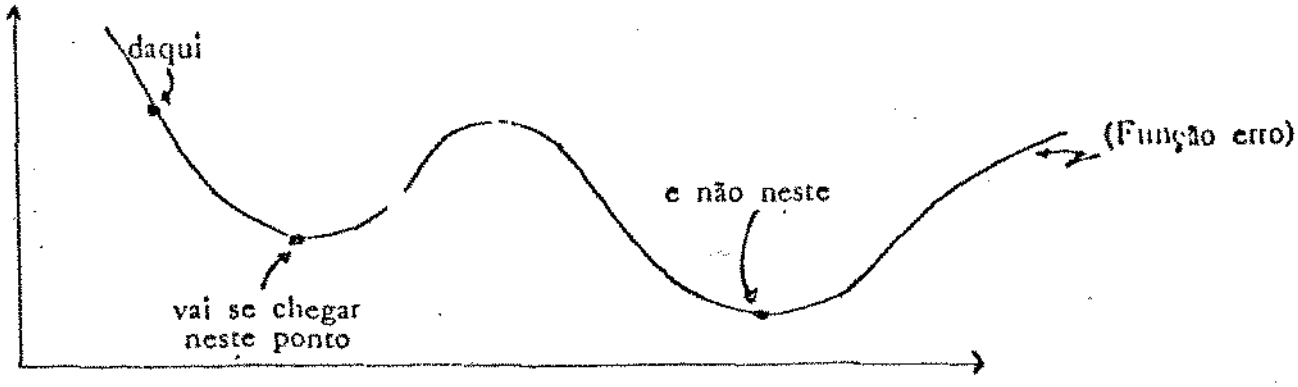
Taxa de Aprendizado

O parâmetro k é conhecido como sendo a *taxa de aprendizado*; quanto menor for k mais lentamente a rede "aprende", porém mais estável é o algorítimo. Como se sabe do Cálculo Integral, para descer uma "ladeira" (uma hiper-superfície), se deve "caminhar" na direção oposta ao vetor gradiente. Porém se o "passo" dado nessa direção for muito grande,

corre-se o risco de "subir" a ladeira do outro lado do "vale". Exemplo:



Devido a isto a taxa de aprendizado deve ser preferivelmente pequena, ainda que o aprendizado seja lento, evitando-se assim "atravessar o vale" . Um outro ponto que deve ser levado em consideração é que o algorítimo não garante o menor valor possível para o erro $E(\vec{w})$. Isto pode ser visto na ilustração abaixo:



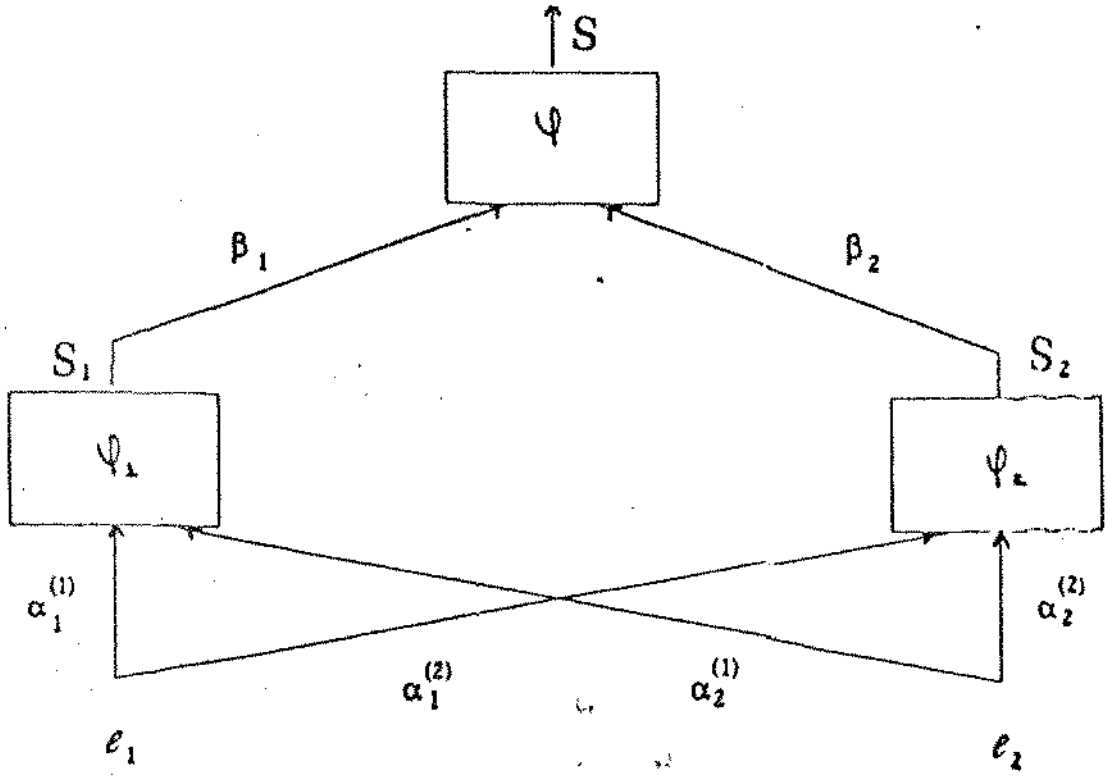
De fato, o mínimo que será obtido depende do ponto de partida; isto é típico dos algorítmos de minimização de funções não lineares. Uma forma de amenizar esta desvantagem é aplicar o algorítmo várias vezes, partindo de pontos iniciais diferentes. Outro é começar com passos bem longos, e aos poucos diminuir os passo, com a esperança que as iterações

fiquem "presas" num mínimo local bem baixo. A seguir, ilustraremos o processo de propagação reversa.

4.7.3 Ilustração Analítica da Dinâmica do Processo de Propagação Inversa do Erro

Vamos desenvolver nesta seção o processo de treinamento de uma rede neural artificial que resolva um problema de duas entradas e uma saída utilizando-se de duas unidades na camada intermediária. A título de ilustração do processo, desenvolveremos o algorítmo utilizado *passagem por passagem* desde suas definições até o processo de atualização de pesos e limiares, a fim de podermos observar a matemática envolvida no processo, uma simples aplicação do Cálculo Diferencial.

Desenvolvimento Analítico



Esta é a representação de uma rede que possui duas entradas e_1, e_2, uma camada escondida de dois elementos cujas saídas são s_1 e s_2 e finalmente uma saída s. Os pesos associados às conexões bem quanto os limiares são mostrados no esquema acima. Vejamos agora as equações da rede, supondo que as funções de ativação das unidades sejam sigmoides monopolares:

$$s_1 = \frac{1}{1 + \exp(\alpha_1^{(1)} e_1 + \alpha_2^{(1)} e_2 - \theta_1)}$$

$$s_2 = \frac{1}{1 + \exp(\alpha_1^{(2)} e_1 + \alpha_2^{(2)} e_2 - \theta_2)}$$

$$S = \frac{1}{1 + \exp(\beta_1 s_1 + \beta_2 s_2 - \theta)}$$

Seja agora dado um conjunto de treinamento

$$(e_1^{(1)}, e_2^{(1)}, s^{(1)}), \ldots, (e_1^{(p)}, e_2^{(p)}, s^{(p)}) \text{ de } p \text{ padrões.}$$

Para "treinar" a rede, devemos minimizar a função de erro:

$$E^2(\alpha_1^{(1)}, \alpha_1^{(2)}, \alpha_2^{(1)}, \alpha_2^{(2)}, \theta_1, \theta_2, \beta_1, \beta_2, \theta) =$$
$$\sum_{j=1}^{p} (S(e_1^{(j)}, e_2^{(j)}) - S^{(j)})^2$$

Como este é um problema de minimização não linear, utilizaremos o método de decaimento por gradiente. Isto não quer dizer que outros métodos não funcionem; podem ser também usados métodos de penalidade, ou resolver o sistema não linear das derivadas parciais por método tipo Newton. Porém, em redes com um número grande de componentes, a tarefa computacional é imensa; devido a isto o método do gradiente é preferido em geral.

Simplificando a notação usando

$$(w_1, \ldots, w_9) = (\alpha_1^{(1)}, \alpha_1^{(2)}, \alpha_2^{(1)}, \alpha_2^{(2)}, \theta_1, \theta_2, \beta_1, \beta_2, var)$$

e começando com um certo $(w_1^{(0)}, \ldots, w_9^{(0)}) = \vec{w}^{(0)}$ teremos que efetuar as iterações:

$$\vec{w}^{(t+1)} = \vec{w}^{(t)} - k \text{ grad } E(\vec{w}^{(t)})$$

com a taxa de aprendizado $0 < k < 1$. Para uma componente w_i de $\vec{w}$ tem-se que:

$$w_i^{(t+1)} = w_i^{(t)} - k \frac{\partial E(\vec{w})}{\partial w_i}\bigg|_{\vec{w}^{(t)}}$$

e esta atualização de $w_i^{(t+1)}$ deve ser feita para cada uma das unidades $i = 1, \ldots, q$.

Retro-Propagação Global x Local

O esquema iterativo acima é um esquema de propagação reversa global, já que usa todo o conjunto treino para atualizar $\vec{w}_i$. Na prática, usa-se propagação reversa local, o que é feito da seguinte forma:

Definimos

$$\begin{aligned} E_j^2(\vec{w}) &= (S(e_1^{(j)}, e_2^{(j)}) - S^{(j)})^2 \\ E(\vec{w}) &= \sum_{j=1}^{p} E_j^2(\vec{w}) \end{aligned}$$

Assim, no esquema global

$$w_i^{(t+1)} = w_i^{(t)} - k \sum_{j=1}^{p} \frac{\partial E_j^2(\vec{w})}{\partial w_i}\bigg|_{\vec{w}^{(t)}}$$

ou em pseudo código, a iteração n é:

Para $i = 1, 2, \ldots, q$

Faça:

$\Big\{$Para $j = 1, 2, \ldots, p$

Faça:

$$w_i^{(t+1)} = w_i^{(t)} - k \frac{\partial E_j^2(\vec{w})}{\partial w_i}\bigg|_{\vec{w}^{(t)}} \Big\}$$

No esquema local usa-se um elemento de treino por vez, e não todos eles, isto é, a iteração sobre i se faz dentro da iteração sobre j:

Para $j = 1, \ldots, p$

Faça:

$$\left\{ \begin{array}{l} \text{Para } i = 1, 2, \ldots, q \\ \text{Faça:} \\ w_i^{(t+1)} = w_i^{(t)} - k \left. \dfrac{\partial E_j^2(\vec{w})}{\partial w_i} \right|_{\vec{w}(t)} \end{array} \right\}$$

A idéia é que a correção no valor de $\vec{w}$ deve ser feita apresentando somente uma "situação" (elemento do conjunto de treino) e corrigindo $\vec{w}$ de acordo com a resposta da rede a esta situação específica; logo se apresenta uma outra situação e se corrige $\vec{w}$, e assim por diante. No esquema global as correções de $\vec{w}$ são feitas sobre a média das respostas da rede a todas as situações. Em analogia com o aprendizado, isto pode ser entendido da seguinte forma : Suponha que para um grupo de alunos se dá um teste de 10 perguntas, e ao final do teste cada aluno fica com uma cópia do teste e das respostas que ele deu às 10 perguntas. No esquema global, o aluno somente recebe a nota do teste; isto é, a média das notas que tirou nas 10 perguntas. Já no esquema local, o aluno recebe as 10 notas, e não somente a média. No segundo esquema fica mais fácil para o aluno saber onde ele errou, e onde ele acertou. Com isto fica mais claro para o aluno saber quais são seus pontos fracos e quais são seus pontos fortes; ele pode então estudar mais eficientemente no esquema local que no esquema global.

Cálculo Analítico dos coeficientes para Retro-Propagação

Vamos agora desenvolver analiticamente o algoritmo de aprendizado para cada um dos parâmetros w_i. Observe que o processo fica mais simples calculando-se primeiramente $w_9 = \theta$ e logo para $w_8 = \beta_2$, $w_7 = \beta_1$ e assim até w_1. É por isto que o algoritmo chama-se *propagação inversa do erro* ou *Retro-Propagação* , já que as mudanças nos parâmetros são feitas na direção saída $\longrightarrow$ entrada, e não na direção entrada $\longrightarrow$ saída. Porém, isto não tem significado biológico, e rigorosamente falando, nem matemático. É feito assim por ser a forma mais simples e conveniente. Uma vez que os parâmetros são atualizados por meio da equação de diferença

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - k\Delta w_{ij}(t) \tag{4.10}$$

onde calculamos os termos

$$\Delta w_{ij}(t) = \left.\frac{\partial E^2(\vec{w})}{\partial w_{ij}}\right|_{\vec{w}^{(t)}} \tag{4.11}$$

Por simplicidade de notação não escreveremos o sub-índice j.

$w_9 = \theta$:

$$w_9^{(t+1)} = w_9^{(t)} - k\left.\frac{\partial E^2(\vec{w})}{\partial w_9}\right|_{\vec{w}^{(t)}}$$

$$\left.\frac{\partial E^2(\vec{w})}{\partial w_9}\right|_{\vec{w}^{(t)}} = 2E(\vec{w}^{(t)}).S(\vec{w}^{(t)})(1 - S(\vec{w}^{(t)})).$$

$$\left.\frac{-\partial(\beta_1 s_1 + \beta_2 s_2 - \theta)}{\partial\theta}\right|_{\vec{w}^{(t)}}$$

Vamos chamar:

$$\tau_1(\vec{w}^{(t)}) = 2E(\vec{w}^{(t)})S(\vec{w}^{(t)})(1 - S(\vec{w}^{(t)})).$$
$$\exp(\beta_1^{(t)}s_1 + \beta_2^{(t)}s_2 - \theta^{(t)})$$

Assim:

$$\left.\frac{\partial E^2(\vec{w})}{\partial w_9}\right|_{\vec{w}^{(t)}} = -\tau_1(\vec{w}^{(t)}).$$

então: $\theta^{(t+1)} = \theta^{(t)} + k\tau_1(\vec{w}^{(t)})$

$w_8 = \beta_2$:

$$\left.\frac{\partial E^2(\vec{w})}{\partial w_8}\right|_{\vec{w}^{(t)}} = \tau_1(\vec{w}^{(t)}).\left.\frac{\partial(\beta_1 S_1 + \beta_2 S_2 - \theta)}{\partial \beta_2}\right|_{\vec{w}^{(t)}}$$
$$= \tau_1(\vec{w}^{(t)}).S_2(\vec{w}^{(t)})$$

$$\beta_2^{(t+1)} = \beta_2^{(t)} - k\tau_1(\vec{w}^{(t)}).S_2(\vec{w}^{(t)})$$

da mesma forma $w_7 = \beta_1$:

$\beta_1^{(t+1)} = \beta_1^{(t)} - k\tau_1(\vec{w}^{(t)}).S_1(\vec{w}^{(t)})$

$w_6 = \theta_2$:

$\left.\frac{\partial E^2(\vec{w})}{\partial w_6}\right|_{\vec{w}^{(t)}} = \tau_1(\vec{w}^{(t)}).\beta_2^{(t)}\left.\frac{\partial S_2(\vec{w})}{\partial \theta_2}\right|_{\vec{w}^{(t)}}$

agora o processo da derivada da sigmoide se repete para S_2:

$$\left.\frac{\partial S_2(\vec{w})}{\theta_2}\right|_{\vec{w}^{(t)}} = S_2(\vec{w}^{(t)})(1 - S_2(\vec{w}^{(t)})).$$
$$\left.\frac{\partial(\alpha_1^{(2)}e_1 + \alpha_2^{(2)}e_2 - \theta_2)}{\partial \theta_2}\right|_{\vec{w}^{(t)}}$$

Vamos definir:

$$\sigma_2(\vec{w}^{(t)}) = \tau_1(\vec{w}^{(t)})\beta_2^{(t)}S_2(\vec{w}^{(t)})(1 - S_2(\vec{w}^{(t)}))$$

então

$$\left.\frac{\partial E^2(\vec{w})}{\partial w_6}\right|_{\vec{w}^{(t)}} = -\sigma_2(\vec{w}^{(t)})$$

logo

$$\theta_2^{(t+1)} = \theta_2^{(t)} + k\sigma_2(\vec{w}^{(t)})$$

$w_5 = \theta_1$:

Da mesma forma anterior, definindo agora:

$$\sigma_1(\vec{w}^{(t)}) = \tau_1(\vec{w}^{(t)})\beta_1^{(t)}S_1(\vec{w}^{(t)})(1 - S_1(\vec{w}^{(t)}))$$

teremos

$$\theta_1^{(t+1)} = \theta_1^{(t)} + k\sigma_1(\vec{w}^{(t)})$$

$w_4 = \alpha_2^{(2)}$:

$$\begin{aligned}\left.\frac{\partial E^2(\vec{w})}{\partial w_4}\right|_{\vec{w}^{(t)}} &= \sigma_2(\vec{w}^{(t)})\left.\frac{\partial(\alpha_1^{(2)}e_1 + \alpha_2^{(2)}e_2 - \theta_2)}{\partial\alpha_2^{(2)}}\right|_{\vec{w}^{(t)}} \\ &= \sigma_2(\vec{w}^{(t)}).e_2\end{aligned}$$

assim,

$$\alpha_2^{(2)(t+1)} = \alpha_2^{(2)(t)} - k\sigma_2(\vec{w}^{(t)})e_2$$

$w_3 = \alpha_2^{(1)}$:

$$\alpha_2^{(1)(t+1)} = \alpha_2^{(1)(t)} - k\sigma_1(\vec{w}^{(t)})e_2$$

$w_2 = \alpha_1^{(2)}$:

$$\alpha_1^{(2)(t+1)} = \alpha_1^{(2)(t)} - k\sigma_2(\vec{w}^{(t)})e_1$$

e finalmente

$w_1 = \alpha_1^{(1)}$:

$$\alpha_1^{(1)(t+1)} = \alpha_1^{(1)(t)} - k\sigma_1(\vec{w}^{(t)})e_1$$

Resumo do Processo

Em resumo, a cada iteração devemos para cada elemento do conjunto de treinamento inicialmente calcular as saídas das unidades para cada padrão dado

$$S_1(\vec{w}^{(t)}) = \frac{1}{1+\exp(\alpha_1^{(1)(t)}e_1^{(j)} + \alpha_2^{(1)(t)}e_2^{(j)} - \theta_1^{(t)})}$$
$$S_2(\vec{w}^{(t)}) = \frac{1}{1+\exp(\alpha_1^{(2)(t)}e_1^{(j)} + \alpha_2^{(2)(t)}e_2^{(j)} - \theta_2^{(t)})}$$

onde $e_1^{(j)}$ e $e_2^{(j)}$ são as entradas dadas pelo elemento j do conjunto de treino.

$$S(\vec{w}^{(t)}) = \frac{1}{1+\exp(\beta_1^{(t)}S_1 + \beta_2^{(t)}S_2 - \theta^{(t)})}$$

Logo, obtemos o erro local

$$E(\vec{w}^{(t)}) = (S(\vec{w}^{(t)}) - S^j)$$

onde s^j é a saída desejada, isto é, a saída do elemento j do conjunto de treino.

Com isto, atualizam-se os pesos $\vec{w}^{(t)}$ da forma:

$$\theta^{(t+1)} = \theta^{(t)} + k\tau_1(\vec{w}^{(t)})$$

$$\beta_2^{(t+1)} = \beta_2^{(t)} - k\tau_1(\vec{w}^{(t)})S_2(\vec{w}^{(t)})$$

$$\beta_1^{(t+1)} = \beta_1^{(t)} - k\tau_1(\vec{w}^{(t)})S_1(\vec{w}^{(t)})$$

$$\theta_2^{(t+1)} = \theta_2^{(t)} + k\sigma_2(\vec{w}^{(t)})$$

$$\theta_1^{(t+1)} = \theta_1^{(t)} + k\sigma_1(\vec{w}^{(t)})$$

$$\alpha_2^{(2)(t+1)} = \alpha_2^{(2)(t)} - k\sigma_2(\vec{w}^{(t)})e^{(j)}$$

$$\alpha_2^{(1)(t+1)} = \alpha_2^{(1)(t)} - k\sigma_1(\vec{w}^{(t)})e^{(j)}$$

$$\alpha_1^{(2)(t+1)} = \alpha_1^{(2)(t)} - k\sigma_2(\vec{w}^{(t)})e^{(j)}$$

$$\alpha_1^{(1)(t+1)} = \alpha_1^{(1)(t)} - k\sigma_1(\vec{w}^{(t)})e^{(j)}$$

com as funções auxilares τ_1, σ_1 e σ_2 da forma:

$$\tau_1(\vec{w}^{(t)}) = 2E(\vec{w}^{(t)})S(\vec{w}^{(t)})(1 - S(\vec{w}^{(t)}).(-1)$$

$$\sigma_2(\vec{w}^{(t)}) = \tau_1(\vec{w}^{(t)})\beta_2^{(t)}S_2(\vec{w}^{(t)})(1 - S_2(\vec{w}^{(t)}))$$

$$\sigma_1(\vec{w}^{(t)}) = \tau_1(\vec{w}^{(t)})\beta_1^{(t)}S_1(\vec{w}^{(t)})(1 - S_1(\vec{w}^{(t)}))$$

até que se atinja a precisão desejada (observando o erro como critério de parada).

Comentários

Obviamente, o método descrito resolve o problema de discriminação da circunferência enunciado no início desta seção , porém é fácil ver que ele também resolve a cálculo de *qualquer conectivo boolcano binário*, incluindo o **XOR**.

Todavia, é necessário esclarecer que não mais precisamos desenvolver analíticamente o algorítmo para toda rede com topologia diferente desta. Veremos na próxima seção o algorítmo genérico para redes de unidades com uma camada oculta e função de ativação sigmoidal. O estudo analítico particular foi desenvolvido apenas a título de exemplo, embora historicamente até a divulgação do algorítmo de retro-propagação isto fosse feito desta forma.

É necessário ressaltar que podemos observar claramente através deste exemplo de que não há nada além de Cálculo Diferencial elementar no algorítmo de treinamento das redes neurais artificiais por retro-propagação .æ

4.8 O Algorítmo de Retro-Propagação

Nas seções anteriores fornecemos as idéias básicas, os principais mecanismos de treinamento e sobretudo exemplos, inclusive analíticos. Contudo, não foi fornecido um algorítmo genérico nem quais as propriedades matemáticas utilizadas em seu desenvolvimento. Tal coisa o faremos nesta seção .

A divulgação do algorítmo de Retro-Propagação deve-se à Rumelhart, Mc Clelland et all (1986), em seu livro *Paralel and Distribuited Processing* (PDP), uma vez que a primazia do mesmo é, até hoje, objeto de controvérsias.[1] Todavia, fato é que sua utilização em grande escala e popularização deu-se apenas após o advento da publicação do PDP.

Forneceremos aqui o algorítmo de treinamento para redes com uma camada oculta, que todavia pode ser estendido sem dificuldade para redes de mais de uma camada oculta. Não deixaremos, porém, de observar outros modelos e comentários relevantes de arquiteturas interessantes quando houver possibilidade. Um exemplo disto é que no fim desta seção mostramos o surgimento "natural" da *Lei de Hebb*, a partir do modelo matemático apropriado e quais as condições matematicas exigidas para tal.

Embora os resultados aqui apresentados não constituam-se em nenhuma novidade, uma vez que podem ser facilmente encontrados na literatura, embora seja usual apenas referenciar-se àquela apresentada por *Rume-*

[1] Afirma-se que o algorítmo foi desenvolvido independentemente diversas vezes por Bryson e Ho (1969), Werbos (1974), Parker (1985) e Rumelhart, Hinton e Williams (1986) (Kosko,1992).

lhart, Hinton e Williams no capítulo 8 do PDP; visamos aqui suplantar dois leves problemas formais daquela: ser mais concisos e mais compreensíveis, se isto for possível.

Considerações Formais

O Algorítmo de Retro-Propagação é central a grande parte dos trabalhos correntes em redes neurais. O algorítmo perscreve a atualização dos pesos das conexões sinápticas de redes neurais artificiais com camadas ocultas, cujas unidades possuam funções de ativação contínuas e diferenciáveis para que aprendam as relações que governem um conjunto de pares de entrada-saída $(\vec{e}, \vec{s})$. A atualização das conexão sinaptica $w_{ij}(t)$ na $t-$ésima iteração será feita através da equação de diferenças de primeira ordem

$$w_{ij}(t+1) = w_{ij}(t) + k\Delta w_{ij}(t) \tag{4.12}$$

sendo que os neurônios i e j devem ser de camadas contíguas e k é uma constante positiva.

Mais ainda, temos de observar as condições do problema genérico de aprendizado supervisionado, como enunciado na seção 4.6.1.

Portanto, o objetivo do aprendizado é encontrar uma função $F : Q^n \rightarrow S^m$ que realize uma determinada tarefa. Como os conjuntos Q^n e S^m possuem infinitos representantes, devemos tomar uma amostragem finita, constituída de, por exemplo, m pontos significativos (e_i, d_j), onde $e = (e_1, \ldots, e_i, \ldots, e_n) \in Q^n$ será um vetor de entradas e $d =$

$(d_1, \ldots, d_j, \ldots, d_m) \in S^m$ um vetor de saídas desejadas, que sejam amostras significativas dos conjuntos representados. Como na prática o que temos em geral é este conjunto de m amostras, podemos tomar entradas por amostragem com reposição (treinamento por padrão) ou simplesmente tomar ciclos repetidamente sobre o conjunto (treinamento por ciclo).

O estado da rede neural na iteração t (R_t) é definido pela função vetorial $R_t : Q^n \rightarrow S^m$. Por simplicidade de notação , omitiremos o indíce t, denotando a função estado da rede simplemente por R ao invés de R_t. A saída corrente da rede $R(e)$ corresponde ao vetor sinal das saídas $S(y)$:

$$R(e) = S(y) \tag{4.13}$$

$$= (S_1(y_1), \ldots, S_m(y_m)) \tag{4.14}$$

onde abreviamos a função sinal monótona não decrescente S_j^Y por S_j. Como comentamos anteriormente as funções de ativação para o algorítmo de Retro-Propagação devem ser funções diferenciáveis com respeito às suas ativações .

Consideremos o vetor de *erro-quadrático instantâneo* da amostra (e_t, d_t), r_t:

$$r_t = d_t - R(e_t) \tag{4.15}$$

$$= d_t - S(y_t) \tag{4.16}$$

$$= (d_1 - S_1(y_1^t), \ldots, d_m - S_m(y_m^t)) \tag{4.17}$$

o qual é diretamente generalizado pelo método do Minimos Quadrados escalar. O algorítmo de Retro-Propagação usa a soma dos erros instantâneos E_t:

$$E_t = \frac{1}{2}\sum_{j}^{p}[d_j^t - S_j(y_j^t)]^2 = \frac{1}{2}r_t r_t^T \tag{4.18}$$

O **erro total** ou cumulativo E

$$E = \sum_{l} E_t^l \tag{4.19}$$

soma os primeiros l erros instantâneos na iteração t. Em geral,

$$\frac{\partial E}{\partial w_{ij}} = \sum_{t} \frac{\partial E_t}{\partial w_{ij}} \tag{4.20}$$

A ativação do j–ésimo neurônio da camada de saída C_S de uma rede de neurônios formais com uma camada oculta, cuja topologia pode ser representada por $C_E \rightarrow C_O \rightarrow C_S$, é dada por

$$y_j^t = \sum_{q} S_q(a_q^t) w_{qj} \tag{4.21}$$

onde a ativação a_q^t é dada por

$$a_q^t = \sum_{i=1}^{n} e_i w_{iq} \tag{4.22}$$

sendo que o neurônio q pertence a camada oculta C_O.

Lembrando que $S(e_i) = e_i$ para neurônios da camada de entrada em redes de Retro-Propagação .

Derivaremos o algorítmo de Retro-Propagação com repetidas aplicações da regra da cadeia do Cálculo Diferencial em várias variáveis.

4.8.1 Dedução Formal do Algorítmo de Retro-Propagação

Enunciaremos e demonstraremos formalmente agora o algorítmo de Retro-Propagação .

O algorítmo de Retro-Propagação "propaga" o erro quadrático instantâneo, $E_t = \frac{1}{2} r_t^T r_t$, desde a camada de saída C_S, pela camada oculta C_O, até a camada de entrada C_E a cada iteração . As entradas e_t geram erros quadráticos instantâneos. e_t é passada de C_E para C_O. Daí o sinal de C_O é passado para C_S. Então subtraimos o atual vetor de saída $S(y_t)$ do vetor desejado d_t para computar o vetor erro r_t. Repetimos o processo até termos usado todas as amostras do conjunto de treino (e_t, d_t) e, idealmente, alcançado um mínimo local da desconhecida superfície de mínimos quadrados.

O Algorítmo de Retro-Propagação

O Algorítmo de Retro-Propagação consiste em duas fases:

1. Calculam-se as saídas de cada unidade da rede pela apresentação dos padrões do conjunto de treinamento
2. Propaga-se então o erro cometido a partir das unidades da camada de saída para as camadas anteriores, através da regra de atualização das conexões sinápticas

O cálculo das saídas das unidades se procedem de forma usual. Observemos agora as regras de atualização das conexões sinápticas.

Atualização das Conexões Sinápticas

1. Se o $j-$ésimo neurônio pertence à camada de saída C_S, então atualizamos os pesos de suas conexões através de

$$\Delta w_{qj}(t) = [d_j^t - S_j(y_j^t)]S_j'(y_j^t)S_q(a_q^t) \tag{4.23}$$

2. Se o $q-$ésimo neurônio pertence á única camada oculta C_O de uma rede, então

$$\Delta w_{iq}(t) = -[\sum_j^p \frac{\partial E_t}{\partial y_j^t} w_{qj}]S_q'(a_q^t)e_i^t \tag{4.24}$$

$$= [\sum_j^p [d_j^t - S_j(y_j^t)]S_j'(y_j^t)w_{qj}]S_q'(a_q^t)e_i \tag{4.25}$$

Demonstração : Derivaremos o algorítmo de Retro-Propagação pela aplicação sucessiva da regra da cadeia

$$\frac{\partial f}{\partial y_j} = \sum_i^n \frac{\partial f}{\partial e_i}\frac{\partial e_i}{\partial y_j} \tag{4.26}$$

onde $f = f(e_1, \ldots, e_n)$ e $e_i = e_i(y_1, \ldots, y_m)$.

- Suponhamos que o $j-$ésimo neurônio pertença a camada de saída C_S. Então

$$\Delta w_{qj}(t) = -\frac{\partial E_t}{\partial w_{qj}} \tag{4.27}$$

$$= -\frac{\partial E_t}{\partial y_j^t}\frac{\partial y_j^t}{\partial w_{qj}} \quad (4.28)$$

$$= -\frac{\partial E_t}{\partial y_j^t} S_q(a_q^t) \quad (4.29)$$

como $\frac{\partial y_j^t}{\partial w_{qj}} = S_q(a_q^t)$ (pela derivação de (4.21)), temos que

$$\Delta w_{qj}(t) = -\frac{\partial E_t}{\partial S_j(y_j^t)}\frac{\partial S_j(y_j^t)}{\partial w_{qj}} S_q(a_q^t) \quad (4.30)$$

$$= -\frac{\partial E_t}{\partial S_j(y_j^t)} S_j'(y_j^t) S_q(a_q^t) \quad (4.31)$$

mas como S_j é a função de ativação e $S_j' = dS_j\ dy_j$, e pela aplicação da regra da cadeia a (4.18)

$$\frac{\partial E_t}{\partial S_j} = -[d_j^t - S_j(y_j^t)]\frac{\partial S_j}{\partial S_j}$$

$$= -[d_j^t - S_j(y_j^t)]$$

temos que

$$\Delta w_{qj}(t) = [d_j^t - S_j(y_j^t)] S_j'(y_j^t) S_q(a_q^t) \quad (4.32)$$

o que demonstra (4.23).

– Suponha agora que o o q–ésimo neurônio pertence á única camada oculta C_O de uma rede cuja topologia é dada por $C_E \rightarrow C_O \rightarrow C_S$, e que todos os neurônios da camada C_E possuem funções de ativação linear ($S_i(e_i) = e_i$), então

$$\Delta w_{iq}(t) = -\frac{\partial E_t}{\partial w_{iq}} \tag{4.33}$$

$$= -\frac{\partial E_t}{\partial a_q^t}\frac{\partial a_q^t}{\partial w_{iq}} \tag{4.34}$$

$$= -\frac{\partial E_t}{\partial a_q^t}e_i^t \tag{4.35}$$

como $\frac{\partial a_q^t}{\partial w_{iq}} = e_i^t$ (pela derivação de (4.22)), temos que

$$\Delta w_{iq}(t) = -\frac{\partial E_t}{\partial S_q(a_q^t)}\frac{\partial S_q(a_q^t)}{\partial w_{iq}}e_i^t \tag{4.36}$$

$$= -\frac{\partial E_t}{\partial S_q(a_q^t)}S_q'(y_q^t)e_i^t \tag{4.37}$$

$$= -[\sum_j^p \frac{\partial E_t}{\partial y_j^t}\frac{\partial y_j^t}{\partial S_q}]S_q'(a_q^t)e_i^t \tag{4.38}$$

$$= -[\sum_j^p \frac{\partial E_t}{\partial y_j^t}w_{qj}]S_q'(a_q^t)e_i^t \tag{4.39}$$

mas como $\frac{\partial y_j^t}{\partial S_q} = w_{qj}$ e por (4.32) até (4.32), temos que

$$\Delta w_{iq}(t) = [\sum_j^p [d_j^t - S_j(y_j^t)]S_j'(y_j^t)w_{qj}]S_q'(a_q^t)e_i \tag{4.40}$$

o que demonstra (4.25). **c.q.d.** / /

Comentários

A derivação não requer que as funções de ativação neuronal sejam monótonas-decrescentes. Assumimos apenas sua diferenciabilidade. Todavia, na prática, usa-se em geral funções de ativação logistica ($S' = S(1-S)$).

Observe que o algorítmo de Retro-Propagação reduz-se ao Método dos Mínimos Quadrados se todas as unidades forem lineares e se utilizarmos uma topologia de Retro-Propagação da forma $C_E \rightarrow C_S$, sem nenhuma camada oculta (este procedimento também foi conhecido nos anos 60 por *ADALINE*, e implementado por Widrow-Hopf). A equação (4.23) reduz-se a

$$\Delta w_{qj}(t) = [d_j^t - S_j(y_j^t)]\frac{y_j^t}{y_j^t}e_i^t \tag{4.41}$$

$$= r_j^t e_i^t \tag{4.42}$$

que se reduz a equação de diferença ordinária (substituindo-se (4.42) em (4.12), temos

$$w_{ij}(t+1) = w_{ij}(t) + c_t r_j^t e_i^t \tag{4.43}$$

que é exatamente a *Regra de Hebb*. Desta forma, a regra de Hebb surge naturalmente quando tratamos de conexões neuronais lineares e sem camadas intermediárias.æ

4.9 Melhoramento da convergência dos algorítmos de Treinamento

O algorítmo de treinamento apresentado (Propagação reversa do erro) é baseado no decaimento simples pelo gradiente; esta técnica, embora frequentemente utilizada, possui uma série de problemas quanto a sua robustez (existência de minimos locais), de demora na convergência do treinamento (pelo uso de k baixas, etc). Comentaremos nesta seção alternativas e técnicas utilizadas para refinamento dos métodos de treinamento, as quais serão utilizadas neste trabalho.

4.9.1 Momentum

A convergência do treinamento melhora se usarmos uma forma de gradientes conjugados. Isto introduz um termo nas iterações que na literatura é chamado de *"momentum"*. Genericamente, o esquema iterativo desenvolvido para um certo parâmetro ξ é:

$$\xi^{(t+1)} = \xi^{(t)} - k\frac{\partial E^2(\vec{w})}{\partial \xi}\bigg|_{\vec{w}^{(t)}}$$

o termo de momentum é da forma

$$m(\xi^{(t)} - \xi^{(t-1)})$$

com isto, o esquema iterativo fica:

$$\xi^{(t+1)} = \xi^{(t)} - k\frac{\partial E^2(\vec{w})}{\partial \xi}\bigg|_{\vec{w}^{(t)}} + m\left(\xi^{(t)} - \xi^{(t-1)}\right)$$

com $0 < m < 1$ o coeficiente de "inércia" (usualmente, usa-se $m = 0,9$) e $0 < k < 1$ a taxa de aprendizado. Computacionalmente, o termo de

momentum é "barato" (envolve poucos cálculos), já que apenas é o incremento do parâmetro na iteração prévia, multiplicado pelo fator de inércia. Deve-se todavia ter o cuidado de guardar na memória o valor presente e o valor prévio de cada parâmetro e não somente o valor presente. O termo de momentum evita variações muito grandes de uma iteração para outra, particularmente em lugares onde a hiper-superficie $E^2(\vec{w})$ é "muito enrugada" e o gradiente for grande.

4.9.2 Gradiente Unitário

Outra forma de se evitar, algumas vezes, a instabilidade do algoritmo é através da normalização do gradiente. Isto envolve, como primeiro passo, o cálculo do gradiente grad $E^2(\vec{w})|_{\vec{w}(t)}$; o comprimento deste vetor: $|$ grad $E^2(\vec{w})|_{\vec{w}(t)}|$ vai ser um número N não negativo. Se este número for maior que um, (e somente então) muda-se temporariamente a taxa de aprendizado k para $\frac{k}{N}$; após completada a iteração, deve retornar o valor de k inicial de outra forma, a taxa de aprendizado irá diminuir cada vez N for maior que 1, e após algumas iterações será praticamente nula.

4.10 Exemplo de Simulação Numérica: O Problema do XOR

Exibiremos aqui uma simulação numérica referente ao modelo de rede neural com duas entradas e uma saída.

Usaremos aqui duas unidades na camada oculta, de modo a utilizarmos a topologia discutida em detalhes na seção 4.3.3.

Podemos treinar esta rede para reproduzir o comportamento de qualquer conectivo binário, de modo que escolhemos o problema do *Ou-Exclusivo* (**XOR**) como "tarefa" a ser realizada pela nossa rede.

Ou-Exclusivo : Objetivo:

$$S = \begin{cases} 0 & \text{se } (0,0) \text{ ou } (1,1) \\ 1 & \text{se } (1,0) \text{ ou } (0,1) \end{cases}$$

Tabela Verdade do XOR

e_1	e_2	$XOR(e_1, e_2)$
0	0	0
1	0	1
0	1	1
1	1	0

Box 1. O Problema do Ou-exclusivo

O conjunto de treinamento da rede é portanto a tabela verdade do conectivo XOR (como dada no Box 1). Cada padrão p corresponde a uma linha da tabela verdade, onde as entradas da rede serão e_1 e e_2 e a saída desejada s_d para este padrão será $XOR(e_1, e_2)$.

As unidades usarão funções de ativação sigmóidal contínua monopolar.

Usaremos em nossa simulação pesos iniciais aleatórios e a constante de aprendizado será fixa em 0,5. Adotaremos como critério de parada (do procedimento iterativo) que o erro quadrático total da rede seja inferior a 0,04. Isto é:

$$\begin{aligned} \text{SEQ} &= \sum_{i=1}^{p} E^2(\vec{e}_i) \\ &= \sum_{i=1}^{p} (s_5^{(i)} - s_d^{(i)})^2 \end{aligned}$$

onde $s_5^{(i)}$ é a saída calculada pela unidade na camada de saida fornecendo-se as entradas referentes ao $i-$ésimo padrão a ser treinado e $s_d^{(i)}$ é o $i-$ésimo padrão do nosso conjunto de treinamento.

As seguintes tabelas ilustram as saídas das unidades de processamento das camadas oculta e de saída e o erro total cometido a cada 30 iterações

A tabela 4.1 mostra evolução dos pesos e limiares das unidades de processamento a cada 30 iterações durante o processo de treinamento. A última coluna apresenta a soma quadrática do erro cometido na $n-$ésima iteração .

O resultado exibido na tabela anterior vai apenas até a iteração 289, quando a precisão desejada foi alcançada.

Já a tabela 4.2 nos exibe as saídas computadas pela rede após o treinamento. Observe que a saída da rede (s_5) é precisa (qualitativamente correta) mas não (quantitativamente) exata.

Outra observação interessante é notar que a unidade s_3 funciona executando o "OU lógico", bem como a unidade s_4 executa o "E lógico". Analisando-se os pesos e o limiar da unidade s_5 na tabela 4.1, observamos que esta unidade executa a operação lógica $s_3 \vee \neg s_4$, que é o resultado qualitativamente esperado, a despeito do processo numérico desconhecer tais "operações lógicas". Portanto, num abuso de linguagem, podemos dizer que o processo de aprendizado "descobriu" os conectivos apropriados.

Esta análise é casuística, uma vez que em geral não é possível analisar semanticamente os pesos e limiares da rede, mas serve perfeitamente para ilustrar o funcionamento do processo de aprendizado em redes neurais artificiais.æ

iter	w_{13}	w_{23}	θ_3	w_{14}	w_{24}	θ_4	w_{35}	w_{45}	θ_5	SEQ
0	0,43	0,44	-0,27	-0,03	0,03	-0,40	0,27	0,08	0,27	1.0507
30	0,40	0,41	-0,35	-0,07	0,00	-0,47	0,14	-0,02	0,02	1.0024
60	0,41	0,43	-0,32	-0,07	0,00	-0,48	0,11	-0,03	0,03	1.0001
90	0,44	0,46	-0,28	-0,07	-0,01	-0,48	0,12	-0,04	0,05	0.9999
120	0,50	0,52	-0,21	-0,07	-0,01	-0,48	0,17	-0,05	-0,07	0.9997
150	0,62	0,64	-0,10	-0,07	-0,01	-0,48	0,28	-0,07	-0,14	0.9987
180	0,99	1,00	0,12	-0,06	0,00	-0,48	0,60	-0,14	-0,36	0.9896
210	2,17	2,17	0,10	0,04	0,08	-0,43	1,71	-0,44	-1,17	0.9030
240	4,13	4,14	-0,73	0,89	0,92	-0,49	3,79	-1,59	-2,16	0.6579
270	5,38	5,39	-1,53	2,32	2,32	-3,42	5,47	-5,27	-2,57	0.1605
289	5,71	5,72	-2,16	3,17	3,17	-4,82	6,40	-6,96	-2,82	0.0398

Tabela 4.1: Pesos e limiares da rede XOR

e_1	e_2	s_d	s_3	s_4	s_5
0	0	0	0,10	0,00	0,09
0	1	1	0,97	0,16	0,90
1	0	1	0,97	0,16	0,90
1	1	0	0,99	0,82	0,10
		XOR	"OU"	"E"	"XOR"

Tabela 4.2: Saída numérica da rede XOR

4.11 Implementação da Lógica Fuzzy em Redes neurais Artificias

A implementação da Lógica Fuzzy em Redes Neurais Artificiais de Retro-Propagação é uma tentativa de se explorar sua capacidade de aprendizado funcional.

4.11.1 O enunciado do problema teórico

Consideremos uma "espécie" de abordagem funcional[2] na qual desejamos que a tarefa da rede $F(\vec{e},\vec{p})$ seja aprender a relação de implicação composicional

$$F(\vec{e},\vec{p}) = A' \circ I(A,B) \tag{4.44}$$

onde A, A' e B são conjuntos fuzzy que representam proposições da Lógica Fuzzy.

Como desejamos que nosso operador $I(A,B)$ satisfaça as propriedades semânticas apropriadas a GMP, utilizaremos como conjunto de treinamento pares $(A_i,B_i)_j$ onde A_i e B_i são conjuntos fuzzy linguisticamente modificados para atender a propriedade GMP_j.

A representação dos conjuntos fuzzy acontece de forma natural, uma vez que para efeito de inferência prática, utilizamos conjuntos fuzzy discretos. Desta forma basta considerar uma rede cujo número de unidades na camada de entrada seja a cardinalidade da discretização do conjunto

[2]Isto é do ponto de vista lógico, obviamente, uma heresia, pois seria equivalente a considerar a possibilidade de uma tabela verdade com infinitas linhas. Do ponto de vista teórico, isto é possível através de um reticulado no espaço de funções , mas o problema seria então, com a notação lógica usual, algébricamente intratável. *Graças a Deus*, tal abordagem do problema , como veremos, é desnecessária em nosso caso.

fuzzy A que representa a cláusula antecedente da implicação e o número de unidades na camada de saída iguais a cardinalidade da discretização do conjunto fuzzy B que representa a cláusula consequente da implicação . Desta forma, a cada unidade na camada de entrada será associada a $\mu_A(x_i)$ e da camada de saída a $\mu_B(y_j)$, onde os μ's são os graus de pertinência parcial dos elementos dos respectivos conjuntos.

Utilizaremos em nossa arquitetura uma camada intermediária, de modo a poder armazenar as relações funcionais desejadas. O número de unidades nesta camada e suas consequências serão objeto de estudo neste trabalho.

Utilizaremos redes de Retro-Propagação usuais, cujas unidades possuem função de ativação sigmoidal monovalente e o algorítmo de treinamento dos pesos e limiares será o algorítmo de Retro-Propagação visto no capítulo anterior, com um termo de momentum. Todas as unidades de uma camada serão totalmente conectadas às unidades da camada seguinte.

4.11.2 O problema teórico na prática

Considere que desejamos implementar em RNA a regra

$$\text{Se } x \text{ é } A \text{ então } y \text{ é} B \tag{4.45}$$

que é uma expressão condicional fuzzy. Por simplicidade, tomemos sua forma de implicação $A \Rightarrow B$.

Utilizaremos aqui, por exemplo, o esquema clássico de GMP, constituido por GMP1, GMP2a, GMP3a e GMP4a.

Elaboramos então o conjunto de treinamento constituido portanto dos pares de entrada e saída

A	B	(GMP1)
muito A	muito B	(GMP2a)
$\pm$ A	$\pm$ B	(GMP3a)
$\neg$ A	desconhecido	(GMP4a)

e os apresentamos à rede para treinamento.

Após o treinamento, basta oferecermos como entrada o conjunto A' (premissa) e a saída da rede será o consequente B' desejado.

Consideraremos a seguir exemplos numéricos e simulações .

Um problema de exemplo

Consideremos duas variáveis linguísticas ANT e $CONS$, cujos conjuntos fuzzy que a representam são discretos e seus domínios são o intervalo discreto {1,10}. Usaremos como conjunto de termos linguísticos básicos $BAIXO$, $MÉDIO$ e $ALTO$. Suas funções de pertinência serão números fuzzy trapezoidais idealizados de modo a serem compatíveis com seus significados intuitivos.

Farão parte do conjunto de modificadores linguísticos $MUITO^n$ e $\pm^n$, que denotam termos mais e menos específicos, cujas funções de pertinência são obtidas elevado-se seus termos primários à $n+1$ e $1/(n+1)$ potências, respectivamente. Podemos ter também termos resultantes da sobreposição de rótulos linguísticos, taís como $MUITO\ MÉDIO$.

Consideremos, a título de exemplo, que desejamos implementar a regra de inferência "Se x é $BAIXO$ então y é $ALTO$", onde os conjuntos $BAIXO$ e $ALTO$ são como os dados na tabela I.

Portanto, nossos conjuntos de treinamento deverão equivaler aos pares de entrada-saída

Conjuntos utilizados para treinamento

Entrada	Saída
BAIXO	ALTO
MUITO BAIXO	MUITO ALTO
± BAIXO	± ALTO
NÃO BAIXO	DESCONHECIDO

4.11.3 Simulações Computacionais

Ambientes, Linguagem e Softwares Utilizados

As simulações foram efetuadas em diversos ambientes de computador e programas; foram utilizados originalmente programas portáveis escritos em linguagem C e $C++$, cujas execuções foram efetuadas em equipamentos PC compatíveis e em uma estação $SPARC - SUN$ 1+, e com um programa fornecido por Rumelhart e Mc Clelland [22], cujos resultados variaram em precisão e em tempo de execução , mas não variaram qualitativamente quanto a sua resposta. Todas as simulações e os resultados aqui apresentados foram re-executados utilizando um pacote computacional por nós implementado em linguagem *Mathematica*, o *NEURO FUZZY*

Arquitetura da Rede

Tanto a camada de entrada como a de saída foram tomadas com 10 neurônios de modo a representar os conjuntos fuzzy amostrados. Para a camada intermediária foram efetuados testes com várias arquiteturas, que

utilizaram 1,3,5 e 8 unidades na camada intermediária, respectivamente. Para os pesos iniciais foram utilizados números ao acaso, na faixa [-0.5,0.5].

A tolerância aqui desejada foi que a soma dos quadrados dos erros cometidos estivessem abaixo de 0.001 .

Resultados e Testes

Numerosas simulações foram executadas com regras de cláusulas antecedentes simples e múltiplas. Os resultados apresentados nesta seção são típicos dos obtidos em todos os experimentos.

Utilizaremos aqui um número aproximado de iterações fixado, de modo que entendemos aqui por convergência a estabilização do comportamento do erro, ao invés do simples alcance da primeira aproximação , como o considerado por Rumelhart e McClelland [22].

Observemos inicialmente a performance da rede treinada com *GMP* Clássico segundo a regra "Se *BAIXO* então *ALTO*".

Performance da rede para os termos básicos

Entrada	Saída	Saída da rede										*soma*
	Esperada	1	2	3	4	5	6	7	8	9	10	$erros^2$
baixo	alto	0	0	0	0	0	0	0.25	0.5	0.75	1.	-0.057
muito baixo	muito alto	0	0	0	0	0	0	0.062	0.25	0.562	1.	-0.086
± baixo	± alto	0	0	0	0	0	0	0.5	0.707	0.866	1.	-0.0713
não baixo	desconhecido	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	0.150

Performance da rede treinada com *GMP* segundo a regra "Se *BAIXO* então *ALTO*"

Podemos observar que a rede realiza a inferência com precisão (que é arbitrariamente fixada) nos casos para o qual ela foi treinada.

Podemos observar que, como o esperado, as redes com maior número de neurônios na camada intermediária convergiram mais rapidamente.

número de neurônios na camada intermediária	numero médio de padrões apresentados durante o treinamento
8	1000
5	1500
3	5000
1	10000

Performance da duração do treinamento para aprendizado de uma regra simples em diversas arquiteturas

Parece que para algumas condições iniciais a rede com um neurônio na camada intermediária não consegue aprender algumas das relações intuitivas entre as entradas e saídas, não sendo portanto recomendada à inferência.

Observemos a performance da rede em relações para as quais ela não foi treinada, afim de apreciar sua capacidade de generalização . A performance da rede pode ser vista na tabela abaixo

Entrada	Saída Esperada	1	2	3	4	5	6	7	8	9	10	$soma\ erro^2$
$muito^2$ baixo	$muito^2$ alto	0	0	0	0	0	0	0.00390	0.0625	0.316	1.	-0.326
$(\pm)^2$ baixo	$(\pm)^2$ alto	0	0	0	0	0	0	0.707	0.841	0.930	1.	-0.098
médio	desconhecido	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	0.462
muito médio	desconhecido	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.415
alto	desconhecido	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.652

Comportamento da rede para conjuntos distintos do conj. de treinamento.

Após cada ciclo de treinamento, a rede resultante foi testada entrando-se com diversos conjuntos fuzzy. Os resultados exibidos na tabela acima foram obtidos utilizando uma rede com oito neurônios na camada intermediária usando-se uma amostra de conjuntos fuzzy como entrada. A saída aqui exibida foi típica a todos os experimentos realizados, a excessão daqueles com redes com um neurônio na camada intermediária. Como

pode ser visto, a rede é capaz de realizar extrapolações das relações funcionais existentes entre antecedentes e consequentes. Podemos notar que quando a entrada se afasta de *BAIXO*, por exemplo, *MAIS OU MENOS. MÉDIO*, a rede produz saídas mais próximas de *DESCONHECIDO*. Em todos os casos as respostas podem ser consideradas como inferências apropriadas com relação às suas respectivas entradas.

Como comparação , utilizando a regra de inferência composicional com duas de suas traduções mais usuais, como visto anteriormente, o resultado da entrada da premissa "x é *BAIXO*" foram os conjuntos

$$0.33|1+0.33|2+0.33|3+0.33|4+0.33|5+0.33|6+0.33|7+0.5|8+0.75|9+1|10$$

e

$$0.33|1+0.33|2+0.33|3+0.33|4+0.33|5+0.33|6+0.58|7+0.67|8+0.75|9+1|10$$

para as relações $\max-\min$ e *soma − limitada* respectivamente, enquanto a rede aprendeu o conjunto *ALTO* exatamente (com a precisão arbitrariamente especificável). Comparações similares podem ser efetuadas. Em todos os casos testados, usando a regra composicional de inferência original e outras apresentadas no capítulo 2 produziram resultados mais nebulosos que as respectivas redes e falham em capturar relações funcionais entre as entradas e saídas.

Exibimos em seguida o resultado do treinamento de uma rede de inferência para um número reduzido de padrões, isto é, para o caso da relação tradicional de *Modus Ponens*. Neste caso, o mecanismo de inferência realiza relações funcionais entre todas as variações do antece-

dente na clausula consequente. Todavia, quando a entrada se afasta do antecedente,a rede responde com a saída *DESCONHECIDO*, como esperado. Esta é uma situação para a qual a rede foi sub-treinada, mas produz ainda assim respostas apropriadas. Outras cláusulas de antecedente simples foram implementadas exibindo resultados similares.

Entrada	Saída											*soma*
	Esperada	1	2	3	4	5	6	7	8	9	10	$erro^2$
muito baixo	alto	0	0	0	0	0	0	0.25	0.5	0.75	1.	-0.113
$muito^2$ baixo	alto	0	0	0	0	0	0	0.25	0.5	0.75	1.	-0.164
$muito^3$ baixo	alto	0	0	0	0	0	0	0.25	0.5	0.75	1.	-0.203
$muito^4$ baixo	alto	0	0	0	0	0	0	0.25	0.5	0.75	1.	-0.214
± baixo	alto	0	0	0	0	0	0	0.25	0.5	0.75	1.	-0.056
$(\pm)^2$ baixo	alto	0	0	0	0	0	0	0.25	0.5	0.75	1.	-0.047
$(\pm)^3$ baixo	alto	0	0	0	0	0	0	0.25	0.5	0.75	1.	-0.043
médio	desconhecido	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	0.332

Comportamento da rede

Cláusulas disjuntivas no antecedente usualmente requerem que a regra seja dividida em outras duas regras. Um outro conjunto de experimentos foram realizados para determinar se uma rede simples de inferência pode aprender mais de uma regra, como cláusulas disjuntivas no antecedente. Sob a condição de que ambas as clausulas antecedentes (ou consequentes), não apenas podem ser aprendidas regras disjuntivas, mas também podem ser aprendidas regras diferentes por uma rede de inferência simples. A tabela seguinte exibe o resultado de uma rede simples treinada com as regras

"Se x é *BAIXO* então y é *ALTO*"

e

"Se x é *MÉDIO* então y é *MUITO MÉDIO*".

Entrada	Saída											soma
	esperada	1	2	3	4	5	6	7	8	9	10	$erro^2$
muito baixo	$muito^2$ alto	0	0	0	0	0	0	0.004	0.0625	0.316	1.	-0.217
$(\pm)^2$ baixo	$(\pm)^2$ alto	0	0	0	0	0	0	0.707	0.840	0.930	1.	-0.112
muito médio	$muito^2$ médio	0	0	0.004	0.0625	0.316	1.	0.316	0.0625	0.004	0	-0.029
± médio	± médio	0	0	0.5	0.707	0.866	1.	0.866	0.707	0.5	0	0.977
alto	desconhecido	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	0.184
± alto	desconhecido	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	0.153

Comportamento da rede

Uma rede deste tipo provê uma forma natural de resolução de conflitos, desde cada regra seja considerada no treinamento da inferência. A capacidade de armazenamento de mais de uma regra através de uma única estrutura é um benefício adicional. Em todos os casos testados, incluindo várias entradas para as regras exibidas na tabela acima e para outros conjuntos de regras, a rede de inferência produziu resultados próximos aos esperados, incluindo *DESCONHECIDO* quando a entrada se desviava de todas as cláusulas antecedentes.

4.11.4 Conclusões sobre as simulações

Neste capítulo, propusemos o uso de redes neurais artificiais de retro-propagação como meio de implementação da inferência fuzzy. Foi exibido que tais arquiteturas podem ser treinadas de modo a prender e inclusive extrapolar as relações complexas entre os conjuntos fuzzy que representam as cláusulas antecedente e consequente das relações fuzzy. Podem ser ensinadas diferentes regras a uma mesma estrutura, que mostram serem sensitivas a diferentes variações da clausula antecedente. Através desta abordagem, podemos de fato diminuir o número de regras presentes em determinado sistema especialista, sem perda de generalidade, desde que cada regra represente uma relação entre entradas e saídas. Inclusive, as redes neurais de retro-propagação oferecem um meio de computação pa-

ralela simples, de modo a aliviar a carga computacional extremamente pesada, quando do processamento de Lógica Fuzzy e como método efetivo de resolução de conflito entre regras.æ

4.12 Aplicação : Fito-Indicação usando Lógica Fuzzy Neural

Consideremos o mesmo problema da seção anterior e observemos o funcionamento da Lógica Fuzzy Neural a um problema real.

Desejamos treinar a rede para realizar a inferência representada pela regra

$$(S_1 \Rightarrow \text{algo alcalino}) \wedge (S_2 \Rightarrow \text{ácido}) \wedge (S_3 \Rightarrow \text{fortemente ácido}). \quad (4.46)$$

onde S_1, S_2 e S_3 e *algo alcalino, ácido* e *fortemente ácido* são conjuntos fuzzy como definidos no capítulos 2 e 3.

Nossa rede possui 20 unidades na camada de entrada, 12 unidades na camada intermediária e 17 unidades na camada de saída.

Portanto, assumindo que escolhemos as propriedades de GMP Clássico, devemos treinar a rede com os seguintes pares

Predição da acidez do solo por meio de inferência fuzzy neural

$$W = AL_{as}(F(\vec{T}, \vec{p}))$$

T - Entrada	W - Saída
S_1	algo alcalino
S_2	ácido
S_3	fortemente ácido
muito S_1	(algo alcalino)^2
muito S_2	muito ácido)^2
muito S_3	muito fortemente ácido
$\pm S_1$	$\pm$ algo alcalino
$\pm S_2$	$\pm$ ácido
$\pm S_3$	$\pm$ fortemente ácido
$\neg S_1$	indeterminado
$\neg S_2$	indeterminado
$\neg S_3$	indeterminado

Utilizaremos uma única arquitetura para todas as regras, posto que ainda que elas não sejam conflitantes, a rede deverá possuir a capacidade de generalizar a inferência.

Observamos na tabela seguinte o resultado da aproximação linguística das saídas, dados os antecedentes descritos como entradas para a rede.

Predição da acidez do solo por meio de inferência fuzzy neural

$$W = AL_{as}(F(\vec{T}, \vec{p}))$$

T - Premissa	W - Conclusão
S_1	algo alcalino
S_2	ácido
S_3	fortemente ácido
um fragmento empobrecido de S_1	algo ácido
um fragmento empobrecido de S_2	ácido
um fragmento empobrecido de S_3	fortemente ácido
aproximadamente S_1	algo neutro
aproximadamente S_2	ácido
aproximadamente S_3	fortemente ácido
uma forma transacional entre S_1 e S_2	fracamente ácido
uma forma transacional entre S_1 e S_3	ácido
uma forma transacional entre S_2 e S_3	ácido

Os resultados parecem algo mais intuitivos, quando diferem que os obtidos no capítulo 3. Faremos uma comparação entre os mesmos na seção

seguinte.

4.12.1 Discussão dos Resultados

Comparemos os resultados obtidos pelos métodos clássicos de inferência fuzzy (capítulo 3) e utilizando Lógica Fuzzy Neural.

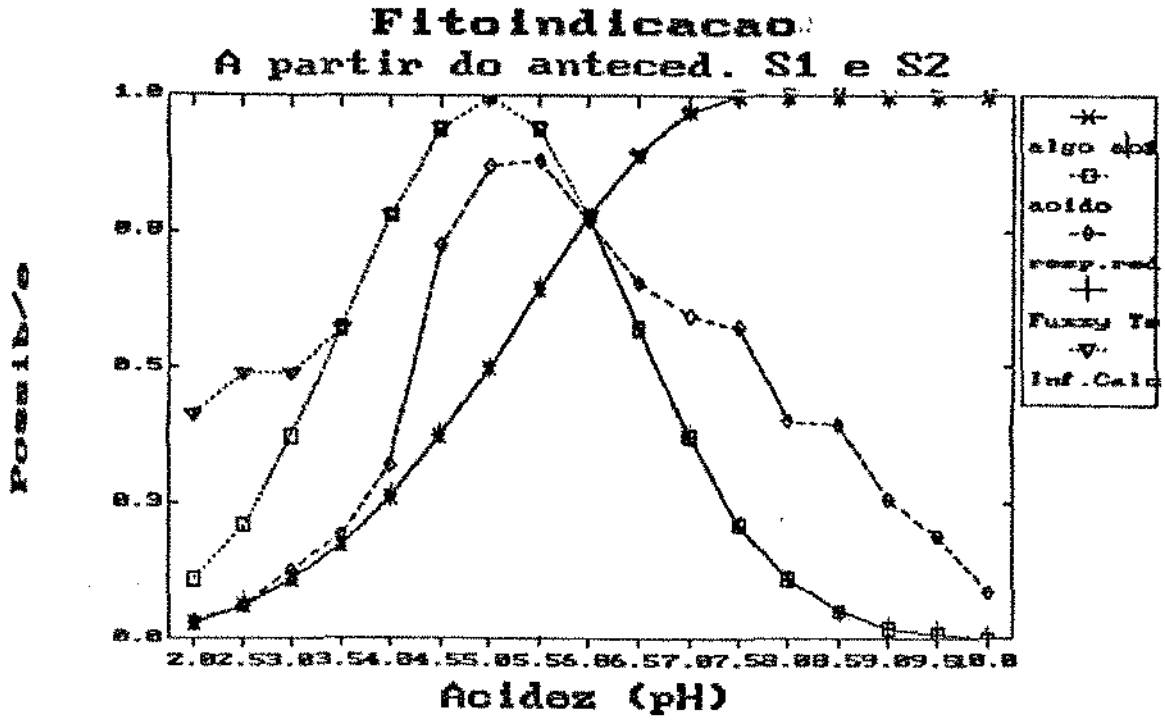
Comparação da predição da acidez do solo por meio de inferência fuzzy neural e de inferência composicional

	Inferência Neural		Inferência Composicional	
T - Premissa	Conclusão	dist	Conclusão	dist
S_1	algo alcalino	0.001	algo alcalino	0.810
S_2	ácido	0.003	ácido	1.009
S_3	fortemente ácido	0.003	fortemente ácido	1.093
um fragmento empobrecido de S_1	algo ácido	0.663	algo alcalino	0.015
um fragmento empobrecido de S_2	ácido	0.742	ácido	0.230
um fragmento empobrecido de S_3	fortemente ácido	0.967	fortemente ácido	0.328
aproximadamente S_1	algo neutro	0.416	algo neutro	0.851
aproximadamente S_2	ácido	0.492	algo ácido	0.729
aproximadamente S_3	fortemente ácido	0.344	algo ácido	0.729
não S_1	indeterminado	0.002	algo ácido	0.395
não S_2	indeterminado	0.001	indeterminado	0.281
não S_3	indeterminado	0.001	indeterminado	0.781
uma forma transacional entre S_1 e S_2	fracamente ácido	0.289	algo neutro	0.994
uma forma transacional entre S_1 e S_3	ácido	0.896	indeterminado	0.742
uma forma transacional entre S_2 e S_3	ácido	0.512	algo ácido	1.056

onde *dist* significa a Distância Euclidiana entre o resultado obtido e o conjunto fuzzy que define o rótulo linguístico.

Ao observar a tabela acima podemos notar alguns fatos.

Enquanto a rede neural é precisa nas inferências para a qual foi treinada (por exemplo em *Modus Ponens*), a inferência composicional chega a apresentar o curioso resultado de ser mais precisa nas premissas mais vagas (como "um fragmento empobrecido de...") do que nas premissas nítidas.


Fitoindicacao
A partir do anteced. S1 e S2
Possib/o
1.0
0.8
0.5
0.3
0.0
2.02.53.03.54.04.55.05.56.06.57.07.58.08.59.09.510.0
Acidez (pH)
algo ab
acido
resp.rel
Fuzzy Te
Inf.Calc

Algumas diferenças qualitativas ocorreram. Vamos analisar um caso mais detalhadamente. No caso de usarmos a premissa "uma forma transacional entre S_1 e S_3" enquanto a rede apresenta a resposta ácido a inferência composicional apresenta indeterminado.

Intuitivamente, deveriamos esperar uma resposta entre o consequente de S_1 e o de S_3, ou seja, entre algo alcalino e fortemente ácido.

Observemos o gráfico dos conjuntos fortemente ácido, algo alcalino e os resultados da inferência neural e composicional.

Enquanto a inferência composicional funciona neste caso como uma espécie de ϵ, pois toma na dúvida o conjunto menos específico de todos, o resultado da rede neural fica *sempre* entre os dois conjuntos correspondentes aos consequentes, uma resposta bastante intuitiva, independentemente do rótulo linguístico a que foi associada. Cabe notar que a rede não foi treinada para este caso específico, pois este antecedente não faz parte do conjunto de treinamento, tendo portanto, generalizado o operador de inferência.

4.13 Conclusão

Outros modelos, como o controle de população de peixes em grandes lagos do nordeste [Blinder e Paes,1993] foram implementados através da mesma técnica, com êxito na validação das hipóteses empregadas.

Vários trabalhos voltados a elaboração de sistemas especialistas em ecologia usando Lógica Fuzzy Neural têm sido elaborados e testados com base no trabalho aqui exposto [Blinder e Bassanezi 1993; Blinder, 1994;

Blinder e Paes, 1994; Blinder, Paes, Sheppard e Bassanezi, 1994].

Atualmente trabalhamos na validação dos modelos fitosociológicos aqui empregados, com os Professores Eduardo Tavares Paes (IO-USP) e George Sheppard (IB-UNICAMP).

Podemos concluir que a representação de problemas biológicos, cuja vaguidão, incerteza e imprecisão linguísticas são inerentes aos métodos e dados, através da Teoria Fuzzy é poderosa e útil. Superando-se as dificuldades técnicas inerentes à inferência fuzzy através de sua implementação em redes neurais artificiais, podemos obter uma ferramenta com grande poder de síntese, possibilitando assim a modelagem, representação e testes de validação de teorias e resultados, extremamente difíceis nos problemas do dia-a-dia real das Ciências Biológicas.æ

Bibliografia

[1] G. Banon. Distinction between several subsets of fuzzy measures. *Fuzzy Sets and Systems*, 5:291–305, 1981.

[2] L.C. Barros. *Modelos Deterministicos com parametros subjetivos.* 1992.

[3] P.B. Blinder R.C. Bassanezi. Modelagem em ecologia teórica através de sistemas especialistas fuzzy. In *XVI CNMAC - Uberlândia*, 1993.

[4] R.C. Bassanezi. *Medidas e Integrais Fuzzy.* Technical Report n.o 06/87, 1987.

[5] P.B. Blinder. Logica fuzzy como fator de síntese e suas implicações em ciencias cognitivas. *Coleção Documentos IEA-USP*, Novembro 1993.

[6] P.B. Blinder. Neural fuzzy logics as a tool for design ecological expert systems. In *Proceedings of WCNN'94*, International Neural Network Society, San Diego, USA, 1994.

[7] R.M. Tong J. Efstathiou. A critical assessment of truth functional modification and its use in approximate reasoning. *Fuzzy Sets and Systems*, 7:103–108, 1982.

[8] A.B. Engel. Uma introdução a redes neurais com aprendizado su-

pervisionado e sinapse não linear. *Biomatemática*, setembro 1992.

[9] D.E. Rumelhart J.L. McClelland et alli. *Parallel Distributed Processing: Explorations in the Microstructure of cognition.* Bradford Book, 1986.

[10] H. Gargantini et alli. *Levantamento de fertilidade do solo do Estado de São Paulo.* Inst.Agronômico de Campinas, 1970.

[11] P.B. Blinder E.T.Paes. An improvement in fuzzy logic for phytosociology. *Vegetatio (submitted).*

[12] J.R. Geronimo. *Medidas Fuzzy.* 1988.

[13] J.M.Keller H.Tahani. Backpopagation neural networks for fuzzy logic. *Information Sciences*, 62, 1992.

[14] I. Graham P.L. Jones. *Expert Systems: Knowledge, Uncertainty and Decision.* Chapman and Hall, 1988.

[15] A.N. Kolmogorov. Dolk. arad. navk. *A.M.S. Translations*, 2(55), 1957.

[16] B. Kosko. *Neural Networks and Fuzzy Systems: A dynamical systems approach to machine intelligence.* Prentice-Hall International, 1992.

[17] M. Mizumoto S. Fukami K.Tanaka. Some methods of fuzzy reasoning. In M.M. Gupta R.K. Ragade R.R.Yager, editor, *Advances in Fuzzy Set Theory and Applications*, North Holland, Amsterdam, 1979.

[18] A.P. Maranca. A common fuzzy logic background to neural networks and linguistic synthesis technique. *Coleção Documentos IEA-USP,*

Outubro 1992.

[19] W. Matuszkiewicz. *A guide for identification of the plant comunities of Poland.* PWN, 1981.

[20] I.R. Moraczewski. Fuzzy logic for phytosociology: 1. sintaxa as a vague concept. *Vegetatio*, 106, 1993.

[21] I.R. Moraczewski. Fuzzy logic for phytosociology: 2. generalizations and predictions. *Vegetatio*, 106, 1993.

[22] J. Moravec. Influences in individualistic concepts of vegetation on syntaxonomy. *Vegetatio*, 81, 1989.

[23] P.B. Blinder E.T. Paes. Regulação da abundância de comunidades de peixes pelo número de predadores utilizando a implementação da lógica fuzzy em redes neurais. In *XVI CNMAC - Uberlândia*, 1993.

[24] D. Dubois H. Prade. *Fuzzy sets and Systems: Theory and Applications.* Academic Press, 1980.

[25] D.W. Roberts. Fuzzy systems vegetation theory. *Vegetatio*, 83, 1989.

[26] D.W. Roberts. Ordination based on fuzzy set theory. *Vegetatio*, 66, 1986.

[27] J.L. McClelland D.E. Rumelhart. *Exploration in Parallel Distributed Processing: A Handbook of Models, Programs and Exercises.* Bradford Book, 1988.

[28] A. Salski. Fuzzy knowledge-based models in ecological research. *Ecological Modelling*, 63, 1992.

[29] T. Whalen B. Schott. Alternative logics for approximate reasoning in expert systems: a comparative study. *Int.J.Man-Machine Studies*,

22, 1985.

[30] M.Minsky S.Papert. *Perceptrons*. Cambridge, 1969.

[31] M. Sugeno. *Theory of Fuzzy Integrals and its Applications*. PhD thesis, Tokyo Institute of Technology, 1974.

[32] R.R. Yager. A measurement-information discussion of fuzzy union and intersection. *Int.J.Man-Machine*, 11:189–200, 1982.

[33] L.A. Zadeh. The concept of linguistic variable and its application to approximate reasoning. *Information Sciences*, 8:199–249,301–357, 1975.

[34] L.A. Zadeh. Fuzzy logic. *IEEE Magazine*, Abril 1988.

[35] L.A. Zadeh. Fuzzy sets. *Information Control*, 8, 1965.

[36] L.A. Zadeh. A theory of approximate reasoning. In e L.I. Mikulich J. Hayes, D. Michie, editor, *Machine Intelligence*, Halstead Press, New York, 1979.

[37] E. Feoli V. Zuccarello. Syntaxonomy: a source of useful fuzzy sets environmental analysis? *Coenoses*, 3, 1988.