

ERIC BASTOS GÖRGENS

**ESTIMAÇÃO DO VOLUME DE ÁRVORES UTILIZANDO  
REDES NEURAIS ARTIFICIAIS**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência Florestal, para obtenção do título de *Magister Scientiae*.

VIÇOSA  
MINAS GERAIS – BRASIL  
2006

**Ficha catalográfica preparada pela Seção de Catalogação e  
Classificação da Biblioteca Central da UFV**

T

G667e  
2006

Görgens, Eric Bastos, 1983-

Estimação do volume de árvores utilizando redes neurais artificiais / Eric Bastos Görgens. – Viçosa : UFV, 2006. viii, 84f. : il. (algumas col.) ; 29cm.

Inclui apêndice.

Orientador: Helio Garcia Leite.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Inclui bibliografia.

1. Árvores - Medição. 2. Redes neurais (Computação).  
3. SNNS (Programa de computador). 4. Florestas -  
Administração. I. Universidade Federal de Viçosa.  
II. Título.

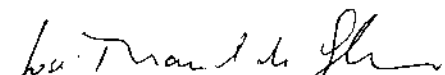
CDD 22.ed. 634.9524

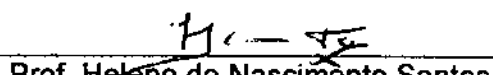
ERIC BASTOS GÖRGENS


**ESTIMAÇÃO DO VOLUME DE ÁRVORES UTILIZANDO  
REDES NEURAS ARTIFICIAIS**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência Florestal, para obtenção do título de *Magister Scientiae*.

APROVADA: 29 de novembro de 2006.

  
Prof. José Marinaldo Gleriani  
(Co-Orientador)

  
Prof. Heleno do Nascimento Santos  
(Co-Orientador)

  
Prof. Carlos Pedro Boechat Soares

  
Prof. Agostinho Lopes de Souza

  
Prof. Heio Garcia Leite  
(Orientador)

*Aos meus pais, Robert e Jacy.*

*Ao meu amor, Pollyanna.*

*A você...*

## **AGRADECIMENTOS**

À Universidade Federal de Viçosa e ao Departamento de Engenharia Florestal, pela oportunidade de realização do Programa de Pós-Graduação em Ciência Florestal.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo apoio financeiro.

Ao professor, orientador e amigo, Helio Garcia Leite, por tudo que ele representa para mim.

Aos co-orientadores e professores, José Marinaldo Gleriani e Heleno do Nascimento Santos, pela orientação, pelas sugestões e pela amizade.

Ao professor e amigo, Carlos Pedro Boechat Soares, pelas contribuições.

Aos amigos e companheiros Rodrigo, Débora, Celso, Fabiano, Flávio, Márcio, Pablo, Omar, Marco Antônio, Dalton, entre outros.

## **BIOGRAFIA**

ERIC BASTOS GÖRGENS, filho de Robert Görgens e Jacy Bastos Görgens, nasceu em 5 de junho de 1983, na cidade de Belo Horizonte, Estado de Minas Gerais.

Em dezembro de 2000, concluiu o ensino médio no Colégio Loyola, em Belo Horizonte-MG.

Em abril de 2001, iniciou o curso de Engenharia Florestal pela Universidade Federal de Viçosa, concluindo-o em julho de 2005.

Recebeu Votos de Louvor pelo desempenho acadêmico durante o curso de graduação.

No período de 2004 a 2005, foi bolsista de iniciação científica pelo CNPq.

Estagiou nas empresas Suzano Papel e Celulose e Celulose Nipo-Brasileira – CENIBRA.

Em agosto de 2005, iniciou o Programa de Pós-Graduação em Ciência Florestal, em nível de Mestrado, na Universidade Federal de Viçosa, defendendo tese em 29 de novembro de 2006.

Em outubro de 2006, ingressou como Engenheiro Júnior na Votorantim Metais – Unidade Aço Florestal.

## SUMÁRIO

	<b>Página</b>
RESUMO .....	vii
ABSTRACT .....	viii
1. INTRODUÇÃO .....	1
1.1. Objetivos.....	4
2. REDES NEURAIIS ARTIFICIAIS .....	5
3. PRÉ-PROCESSAMENTO .....	21
ESTIMAÇÃO DO VOLUME DE ÁRVORES UTILIZANDO REDES NEURAIIS ARTIFICIAIS .....	27
Resumo.....	27
Abstract.....	28
1. Introdução .....	29
2. Material e Métodos .....	30
3. Resultados e Discussão .....	34
4. Conclusões .....	38
5. Referências Bibliográficas.....	39
DETERMINAÇÃO DA ARQUITETURA E DO PRÉ-PROCESSAMENTO ÓTIMOS PARA A ESTIMAÇÃO DE VOLUME DE ÁRVORES POR MEIO DE REDES NEURAIIS ARTIFICIAIS .....	40
Resumo.....	40
Abstract.....	41

	<b>Página</b>
1. Introdução .....	42
2. Material e Métodos .....	43
3. Resultados e Discussão .....	46
4. Conclusões .....	50
5. Referências Bibliográficas.....	51
4. CONCLUSÃO GERAL .....	52
REFERÊNCIAS BIBLIOGRÁFICAS.....	53
APÊNDICES .....	55
APÊNDICE A .....	56
APÊNDICE B – ARQUIVO EXEMPLO PARA ENTRADA DE DADOS NA FORMA MATRICIAL .....	65



## RESUMO

GÖRGENS, Eric Bastos, M.Sc., Universidade Federal de Viçosa, novembro de 2006. **Estimação do volume de árvores utilizando redes neurais artificiais**. Orientador: Helio Garcia Leite. Co-orientadores: Heleno do Nascimento Santos e José Marinaldo Gleriani.

As redes neurais artificiais (RNAs) consistem de uma rede de unidade de processamento interligada por pesos ajustáveis. Cada unidade possui uma função matemática que processa o somatório das entradas do respectivo neurônio, gerando um valor de saída. Este valor é então repassado a todos os neurônios seguintes. Essas redes são conhecidas como redes multicamadas ântero-alimentadas. As RNAs podem ser usadas para diversas aplicações, entre elas a aproximação de funções. A estimação de volume de árvores é comumente empregada nos inventários florestais para determinar o volume de árvores, sem que elas precisem ser abatidas. Utilizando dados de cubagem de diferentes empresas florestais, foram testadas algumas formas de pré-processamento dos dados e arquiteturas de RNAs. Algumas das redes obtidas apresentaram estimativas livres de bias, podendo ser utilizadas para estimação de volume de árvores.

## ABSTRACT

GÖRGENS, Eric Bastos, M.Sc., Universidade Federal de Viçosa, November 2006. **Estimating tree volume using artificial neural nets**. Adviser: Helio Garcia Leite. Co-advisers: Heleno do Nascimento Santos and José Marinaldo Gleriani.

Artificial neural networks (ANNs) consist of a processing unit network interconnected by adjustable weights. Each unit has a mathematical function that processes the sum of inputs of the respective neuron, generating an output value. This value is then passed on to all the other neurons of the next layer. Such nets are known as multi-layer feed forward neural nets. ANNs can have several applications, such as function approximation. Tree volume estimate is commonly used in forest inventories to determine tree volume without the need to fell the trees. Some ANN data pre-processing forms and architecture were tested by using tree scaling from different forest enterprises. Some of the nets obtained presented unbiased estimates, and thus can be used for estimating tree volume.

## 1. INTRODUÇÃO

**Redes neurais artificiais (RNAs).** Rede neural artificial é um conjunto massivo e paralelamente distribuído de neurônios, que possuem em seu interior funções de ativação. Esses neurônios podem ser organizados de diversas maneiras. Cada conexão entre neurônios contém pesos, que consistem no conhecimento que a rede adquiriu durante o treinamento.

As redes neurais surgiram de observações e pesquisas sobre o funcionamento de redes biológicas e seus elementos fundamentais, os neurônios, presentes no cérebro humano.

O trabalho que “inaugurou” a área de redes neurais foi o artigo de McCulloch e Pitts, de 1943, denominado de *A logical calculus of the ideas immanent in nervous activity*.

Como ciência, as redes neurais então incluídas no campo da inteligência artificial, sob o paradigma de aprendizado por meio de exemplos.

No campo florestal, sua abordagem é praticamente inexistente, principalmente no Brasil, salvo na área de processamento de imagens digitais, em que se aplicam redes neurais para classificação e refinamento de imagens. Internacionalmente, as redes neurais artificiais também vêm sendo utilizadas para a inversão de modelos de reflectância para cálculo de índice de área foliar (IAF).

As características que mais chamam a atenção nas RNAs são a capacidade de aprender e a de generalizar informações. Em outras palavras,

as RNAs são capazes de, por meio de um exemplo aprendido, generalizar o conhecimento assimilado para um conjunto de dados desconhecidos. Outra característica interessante das RNAs é a capacidade de extrair características não-explicítas de um conjunto de informações que lhes são fornecidas como exemplos.

A rede executa um mapeamento de entrada-saída de maneira não-paramétrica, uma vez que não é necessário estabelecer pressuposições sobre o modelo estatístico dos dados de entrada.

Uma rede neural deve ser estável o suficiente para não captar alterações momentâneas dos dados, além de ser suficientemente plástica para se adaptar às mudanças ao longo do tempo. Este é o famoso dilema da estabilidade-plasticidade (BRAGA *et al.*, 1998).

Através do aprendizado, a rede neural é capaz de captar as evidências e atribuir respostas aos problemas abordados. Além disso, cada neurônio é potencialmente afetado pela atividade de todos os neurônios da rede, anteriores, ou não, a ele.

Os neurônios formam, portanto, uma rede, na qual as ligações paralelas permitem que certas operações sejam realizadas com maior rapidez. Esta propriedade é fundamental na abordagem de problemas complexos. Permite-se o uso tanto de variáveis discretas, quanto de variáveis contínuas, abrindo assim ainda mais seu campo de utilização.

**Manejo Florestal.** Todo trabalho de manejo florestal é precedido, entre outras matérias, da dendrometria e do inventário. A dendrometria trata do levantamento de informações acerca da floresta, sendo as principais: diâmetro, altura e volume. O método de cubagem usualmente empregado nas empresas é destrutivo, em que a árvore abatida é seccionada e posteriormente cubada.

As árvores são abatidas de modo seletivo e representativo em todas as classes de diâmetro encontradas no povoamento. Por meio de relações halométricas, o volume das árvores não-abatidas pode ser estimado, utilizando equações volumétricas.

Existem diversos modelos volumétricos, sendo o mais utilizado o de Schumacher e Hall (1933), cuja relação funcional diz que o volume é uma função da altura e do diâmetro a 1,30 m do solo (CAMPOS e LEITE, 2006):

$$\text{Ln}Y = \beta_0 + \beta_1\text{LnDAP} + \beta_2\text{LnHt} + \varepsilon$$

em que

Ln = logaritmo neperiano;

Y = variável de interesse (geralmente volume em m<sup>3</sup>/ha);

$\hat{\alpha}_i$  = parâmetros do modelo, para  $i = 0, 1$  e  $2$ ;

DAP = diâmetro a 1,30 m do solo, em cm;

Ht = altura total das árvores, em m; e

$\varepsilon$  = erro aleatório.

As equações ajustadas são posteriormente usadas no inventário florestal, para gerar o volume das árvores das parcelas. Estas parcelas, quando totalizadas, serão usadas para gerar o intervalo de confiança para o volume total do povoamento. No caso de existir um acompanhamento dessas parcelas por mais de três anos consecutivos, modelos de crescimento e produção podem ser ajustados.

Um problema importante ao conduzir inventários florestais é a definição das equações volumétricas apropriadas para cada estrato da floresta, podendo este ser por material genético, espaçamento inicial, regime de corte e idade. As equações são atualizadas anualmente (LEITE e ANDRADE, 2002).

Em muitas ocasiões não é possível gerar equações volumétricas para determinados estratos. Por exemplo, no caso de plantios clonais novos e de pesquisa (testes clonais), sempre existirão alguns estratos onde não é possível abater árvores para ajuste de modelos volumétricos, sendo aplicadas equações de estratos semelhantes (CAMPOS e LEITE, 2006).

O fato é que ao longo dos anos as empresas acumulam grande quantidade de dados de cubagem, incluindo árvores com diferentes características e variações de forma (LEITE *et al.*, 1995). Esses dados podem ser utilizados para implementar um projeto de redes neurais, cujo objetivo final é a estimativa do volume de árvores.

**Dissertação.** Esta dissertação foi dividida em sete partes. A introdução busca situar o leitor diante do tema deste trabalho, além de introduzir alguns termos importantes que foram utilizados ao longo dos capítulos posteriores.

Na seqüência, foi abordada a teoria de redes neurais artificiais. A terceira parte contém informações sobre mineração de dados, com foco na técnica de RNAs.

As partes quatro e cinco constituem artigos científicos nos quais o tema central foi o uso de RNAs como aproximador de função para estimação do volume de árvores. No primeiro artigo, o objetivo foi comprovar a eficiência da técnica de RNAs diante da metodologia tradicional, em que se utiliza o modelo de Schumacher e Hall (1933). No segundo artigo, foi dado enfoque às diferentes arquiteturas e formas de processamento, buscando trazer qual a mais apropriada para solucionar o problema da aproximação de função proposto neste projeto.

Por fim, na sexta parte foram listadas as conclusões gerais sobre a dissertação.

Fechando este trabalho, é apresentado um apêndice com informações sobre o *software* de RNAs utilizado.

## **1.1 Objetivos**

Os objetivos deste trabalho foram:

1. Apresentar a técnica de RNAs e o estado atual da estimação de volumes de árvores.
2. Definir os conceitos principais acerca de RNAs.
3. Consolidar procedimentos de manipulação de banco de dados que proporcionem a geração de conhecimento de qualidade.
4. Demonstrar a aplicação de RNAs como estimador do volume de árvores.
5. Aprofundar a discussão sobre arquitetura de redes e pré-processamento.
6. Apresentar um informativo sobre o *software* utilizado neste trabalho.

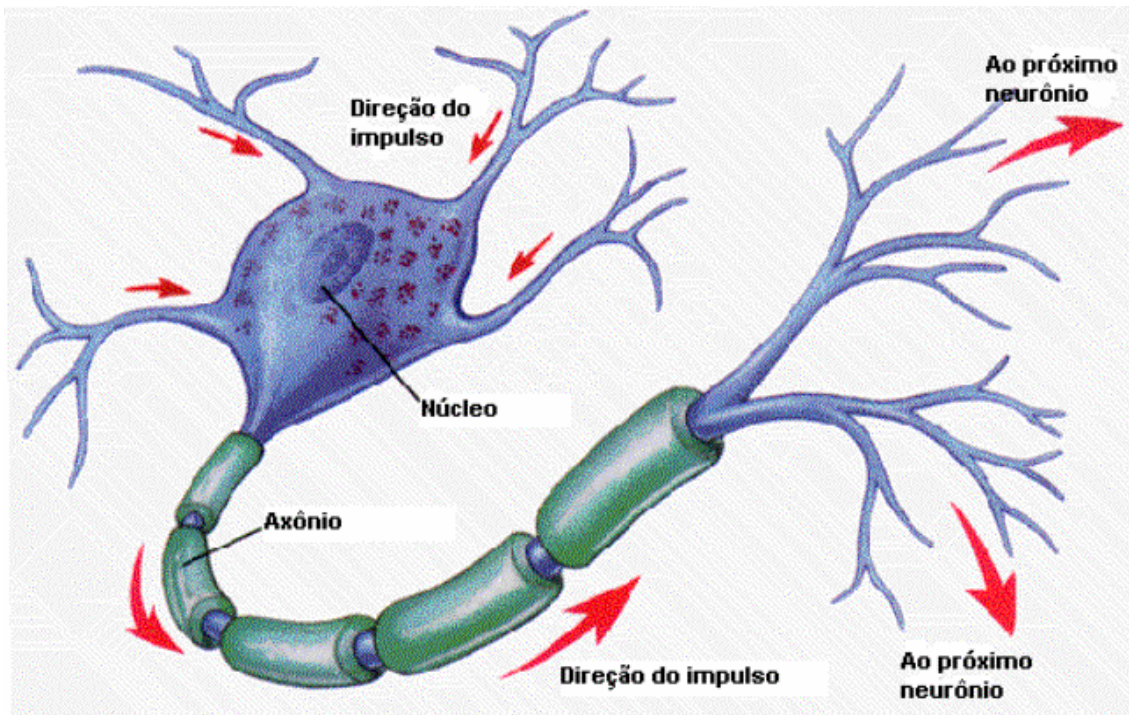
## 2. REDES NEURAIS ARTIFICIAIS

**O neurônio biológico.** Cientificamente, o cérebro humano é o processador de carbono mais fascinante que existe, sendo constituído por cerca de 10 bilhões de neurônios (BRAGA *et al.*, 1998), cabendo a estas tantas unidades o funcionamento de todo o organismo.

Assim, as redes neurais artificiais imitam, ou tentam imitar, o funcionamento das redes neurais biológicas. No entanto, esta semelhança é apenas ideológica, pois na realidade as diferenças são imensas.

Para fundamentar a discussão sobre neurônios e redes neurais artificiais, é necessário conhecer um pouco sobre o neurônio biológico (Figura 1). Este pode ser dividido em três partes, denominadas de dendritos, soma e axônio. Os dendritos são responsáveis pela recepção dos estímulos nervosos. Soma é o corpo da célula, responsável por seus processos metabólicos. Também no corpo celular (soma) o impulso é processado e novos impulsos são gerados, com destino ao exterior da célula, sendo então enviados através do axônio (KOVÁCS, 1996).

Os impulsos químicos podem ser de natureza excitatória ou inibitória, dependendo do tipo de neurotransmissor. Quando a conexão é excitatória, ela trabalha provocando uma alteração no potencial da membrana que favorece a formação de um impulso nervoso. Já a conexão inibitória age no sentido oposto (KOVÁCS, 1996).



Fonte: Brown e Benchmark, *Introductory Psychology Electronic Image Bank*.

Figura 1 – Neurônio biológico.

Assim, presume-se que o sinal de polarização, inibitório ou excitatório, é uma função do tipo de neurotransmissor. Já a magnitude de polarização é uma função da concentração de neurotransmissores, da dimensão da sinapse e do comprimento do neurônio.

Desta maneira, nota-se que todo o conhecimento de uma rede de neurônios biológicos está implícito nas sinapses. Logo, no neurônio artificial, este pressuposto é mantido. As sinapses são representadas pelos pesos, que podem ser positivos ou negativos, e de magnitude diversa.

Em uma rede neural artificial, cada neurônio é uma unidade de processamento composto por uma função de ativação que gera um valor de saída para o neurônio a partir do somatório dos valores que chegam a ele.

**Funções de ativação.** Entre as funções mais comuns estão:

- Função sinal, representada por  $\text{Sgn}(x)$ , é definida por (Figura 2):



$$\text{Sgn}(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{se } x < 0 \end{cases}$$

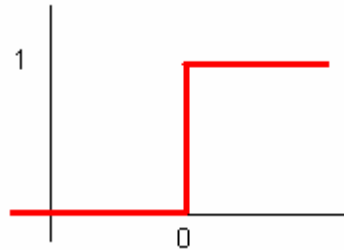


Figura 2 – Função sinal ou degrau unitário.

- Função sigmoial, representada por Sigmoid(x), é definida por (Figura 3):

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

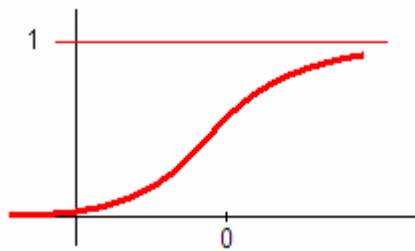


Figura 3 – Função sigmoial.

- Função tangente hiperbólica, representada por Tanh(x/2), é definida por (Figura 4):

$$\text{Tanh}\left(\frac{x}{2}\right) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

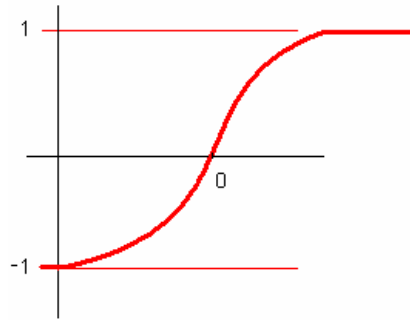


Figura 4 – Função tangente hiperbólica.

- Função *Piecewise* é uma aproximação linear da função sigmoidal, que é definida por (Figura 5):

$$f(x) \begin{cases} 1 & \text{se } x \geq 0,5 \\ x + 0,5 & \text{se } -0,5 \leq x \leq 0,5 \\ 0 & \text{se } x \leq -0,5 \end{cases}$$

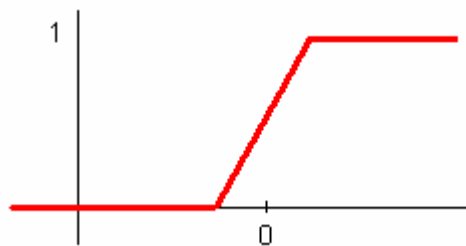


Figura 5 – Função *Piecewise* ou função linear por partes.

**Gradiente e derivadas.** O principal algoritmo de aprendizado das redes neurais, a retropropagação do erro, utiliza o princípio do gradiente para buscar a região no espaço de soluções que leva ao mínimo erro global.

Considere

$$\begin{aligned} f : \mathfrak{R}^n &\rightarrow \mathfrak{R} \\ x &\rightarrow Y = f(x) \end{aligned}$$

diferenciável. O gradiente de  $f$  em  $X$  é o vetor

$$\nabla f(\mathbf{x}) = \left( \frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right)$$

e o método do gradiente para minimização de  $f(\mathbf{x})$  consiste em, a partir de um ponto  $x_k$ , alcançar um ponto  $x_{k+1}$ , sendo  $x_{k+1} = x_k + \eta h_k$ , em que  $\eta$  é o tamanho do passo na direção  $h_k$  e  $h_k = -\nabla f(x_k)$ .

Sabe-se que o gradiente de  $f$  em  $x_k$  aponta a direção de maior crescimento de  $f$ . A direção contrária é a direção de maior decréscimo da função (MATEUS e LUNA, 1986).

**Treinamento.** O processo de treinamento, também conhecido como aprendizado, consiste na modificação ordenada dos pesos, visando a minimização de uma medida de erro, também conhecida como função de custo (HAYKIN, 2001).

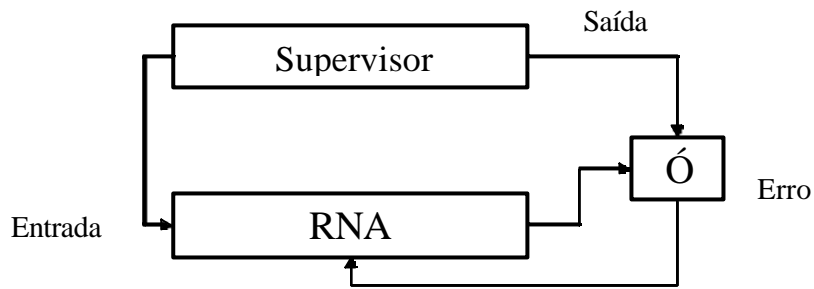
Todo o conhecimento adquirido pela rede fica acumulado nos pesos sinápticos. Analogamente à sinapse do neurônio biológico, o peso é a “memória” da rede.

O processo de treinamento pode ser dividido em três partes bem distintas. Inicialmente, a rede neural é estimulada pelo ambiente. Como resultado deste estímulo, no segundo estágio ocorre a alteração dos seus parâmetros livres. Por fim, a rede neural passa a responder de uma nova maneira ao ambiente que a está estimulando (HAYKIN, 2001).

Como visto em Kovács (1996), Braga *et al.* (1998) e Haykin (2001), o treinamento pode ser dividido em dois grupos: supervisionado e não-supervisionado.

**Treinamento supervisionado.** Conceitualmente, o supervisor possui conhecimento sobre o ambiente e utiliza este conhecimento para indicar à rede qual caminho seguir (Figura 6).

São oferecidos à rede padrões de entrada e de saída. Os dados de entrada são processados pela rede, e a saída atual é comparada com a saída desejada, oriunda do conhecimento prévio do ambiente. Esta comparação é feita por meio de uma medida de erro.

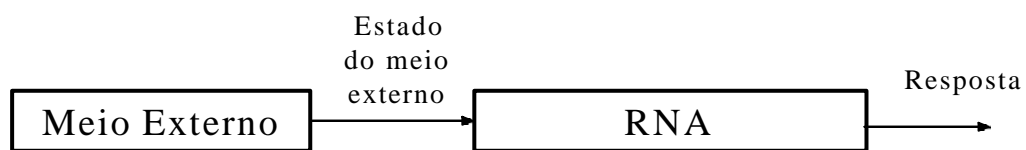


Fonte: adaptado de Haykin (2001).

Figura 6 – Treinamento supervisionado.

O erro é uma função dos parâmetros da rede, podendo assim ser visualizada por uma superfície de erro. Sendo os parâmetros coordenadas dessa superfície, o que se deseja é que o aprendizado leve a rede para o ponto de mínimo global, onde o erro tende a zero.

**Treinamento não-supervisionado.** Neste paradigma de aprendizado, não há conhecimento prévio do ambiente externo. Assim, são oferecidas à rede as condições necessárias para uma análise independente do ambiente. Todo o processo é balizado por essa análise, que é feita tomando-se por base as regularidades apresentadas pelos dados.



Fonte: adaptado de Haykin (2001).

Figura 7 – Aprendizado não-supervisionado.

Diversos algoritmos de aprendizado foram propostos, no entanto a ênfase será dada para apenas três deles: treinamento por correção dos erros, treinamento baseado em memória e treinamento competitivo.

**Treinamento por correção de erros.** Este treinamento é de natureza local, ou seja, os ajustes sinápticos ocorrem em torno de cada neurônio. Essas correções se dão por meio da minimização de uma medida de erro, que nada mais é que a metade do quadrado da diferença entre a saída atual e a desejada. Nesse tipo de aprendizagem, apresentam-se à rede um vetor de entrada e um vetor de saída esperada. Assim, após o vetor de entrada ser processado pela rede, a saída é comparada com a saída desejada através da medida de erro. Esta comparação deve ser, portanto, minimizada (BRAGA *et al.*, 1998; HAYKIN, 2001).

**Treinamento baseado em memória.** Nesta regra de aprendizagem, as experiências são classificadas em uma superfície de resposta. Assim, cria-se uma memória com a qual a rede classifica futuros padrões.

Para essa aprendizagem geralmente utiliza-se o método do vizinho mais próximo ou a função base radial.

Os algoritmos de aprendizagem, com base na memória, se orientam em dois fundamentos. Um critério para dimensionar a vizinhança local e uma regra de aprendizagem aplicada aos dados de treinamento para uma certa vizinhança (HAYKIN, 2001).

**Treinamento competitivo.** Nesta regra de aprendizagem, os neurônios de saída competem entre si para se tornarem ativos. Logo, apenas um neurônio de saída pode estar ativo em um determinado instante. Essa forma de aprendizagem é adequada para inferir sobre características implícitas aos dados, levando a uma autoclassificação do conjunto de padrões de entrada.

Alguns princípios são fundamentais nessa regra de aprendizagem. Inicialmente os neurônios devem ser iguais entre si, exceto pelos pesos sinápticos, que são distribuídos aleatoriamente ao se inicializar a rede.

O segundo princípio é que os neurônios devem estar limitados quanto à força. Por fim, deve existir um mecanismo de competição entre os neurônios, em que aquele que ganha leva o direito de responder.

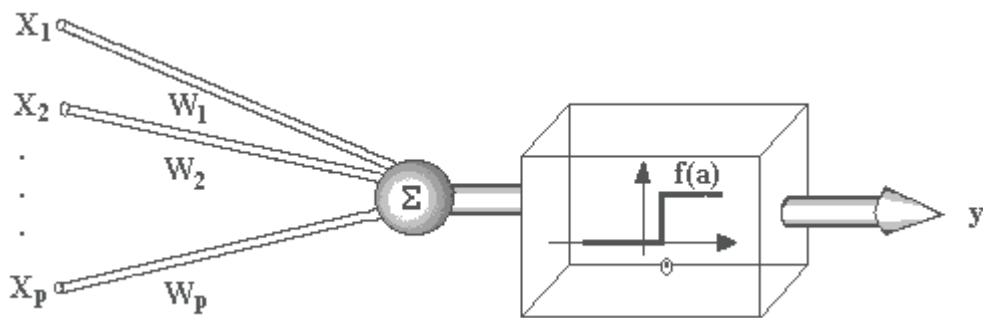
A arquitetura mais simples para esse tipo de treinamento é composta de uma única camada de saída, onde cada neurônio da camada está conectado a

todos os sensores de entrada. Há também ligações laterais entre os neurônios da camada de saída, que funcionam como inibidores laterais.

**As redes neurais.** A primeira tentativa de explicar matematicamente o funcionamento de um neurônio foi feita pelo fisiologista Warren McCulloch e pelo estatístico Walter Pitts, em 1943.

O artigo sob o título *A Logical Calculus of the Ideas Immanent in Nervous Activity* pode ser considerado o marco zero das RNAs.

Basicamente, o neurônio de McCulloch e Pitts, também conhecido como MCP, dirá que  $y = 1$  se  $x \in A$  e  $y = 0$  se  $x \in B$  (Figura 8).



Fonte: disponível em: <<http://www.din.uem.br/ia/neurais/>>.

Figura 8 – Neurônio MCP.

O neurônio MCP é um discriminador linear de entradas binárias, de função de ativação degrau unitário (função sinal).

Logicamente, o neurônio MCP só seria útil nos casos em que os conjuntos a serem separados estivessem aglomerados no espaço  $\mathfrak{R}^n$ , de tal forma que fossem separáveis por um hiperplano.

Em 1958, Rosenblatt demonstrou que se fossem acrescentadas de sinapses ajustáveis, as redes de nodos MCP poderiam ser treinadas (BRAGA *et al.*, 1998). Nota-se que somente em 1958, com Rosenblatt, o termo treinamento foi introduzido.

De modo geral, a rede proposta por Rosenblatt era composta por neurônios MCP como unidade básica e uma regra de aprendizado, o que nada mais é do que uma forma ordenada de atualizar os pesos das conexões da rede neural.

A arquitetura de uma rede *perceptron* é composta por uma camada de entrada, uma camada de associação e uma camada de saída, formada por unidades de resposta. Embora a topologia possua três níveis, ela é conhecida como rede de camada única, uma vez que somente o nível de saída possui características adaptativas. A camada associativa possui pesos fixos, definidos antes do período de treinamento (KOVÁCS, 1996).

No treinamento deseja-se obter o valor  $\Delta w$  a ser aplicado no vetor de pesos  $w$ , para ao final obter uma solução  $w(t+1)$  melhor que  $w(t)$ . O algoritmo de treinamento do *perceptron* sempre chega a uma solução para o problema de duas classes linearmente separáveis em um tempo finito.

Em 1969, Minsk e Papert publicaram um documento de impacto para as redes neurais artificiais. Como consequência dessa publicação, a década de 1970 foi de completo obscurantismo para as redes neurais. Nessa publicação, Minsk e Papert apontavam os defeitos das redes neurais artificiais existentes até então. O principal deles é que as redes, naquela época, só eram eficientes para tratar problemas linearmente separáveis, no entanto a maioria dos problemas reais é de natureza não-linear.

A década obscura (também conhecida como a década perdida) só foi superada com os trabalhos de Hopfield (1982), a respeito de memórias associativas, e com os trabalhos de Rumelhart, Hinton e Williams (1986). Esses últimos foram responsáveis pela divulgação e popularização do algoritmo de treinamento conhecido como retropropagação do erro (*backpropagation*), que permitiu o treinamento de redes com várias camadas.

O problema resolvido pelo algoritmo da retropropagação do erro foi o de como definir o erro dos neurônios das camadas intermediárias.

Cada neurônio de uma rede multicamadas possui uma função de ativação não-linear e diferenciável. Estas características são essenciais para que as mesmas possam ser treinadas. Sendo a função de ativação diferenciável, torna-se possível calcular o gradiente, que é responsável pela indicação da direção de ajuste dos pesos.

**Dificuldades.** O dimensionamento das camadas intermediárias não é uma tarefa de simples resolução. Existem diversos métodos empíricos para dimensioná-la (BULLINARIA, 2003), no entanto nada melhor do que o método da tentativa e erro.

O excesso de neurônios pode levar ao *overfitting* (memorização de ruídos e, conseqüentemente, baixo poder de generalização). Já o baixo número de neurônios pode levar a um custo temporal elevado (às vezes nunca chegando a um resultado ótimo) e, também, pode ser muito influenciado pelos sinais aleatórios dos pesos (HAYKIN, 2001).

Existem alternativas ao dimensionamento empírico. A principal delas diz que após o treinamento da rede aplica-se a técnica de poda (*pruning*), na qual os pesos e neurônios irrelevantes são descartados (BRAGA *et al.*, 1998). A maior crítica que pesa sobre a técnica de poda é a desconsideração da correlação existente entre os neurônios.

Uma alternativa para definir os principais parâmetros de uma RNA é utilizar trabalhos já realizados como referência.

**Retropropagação do erro (*backpropagation*).** Este é o principal algoritmo utilizado para o treinamento de redes multicamadas, utilizando o paradigma do treinamento supervisionado. Por meio de pares de entrada e saída os pesos são ajustados, buscando-se minimizar o erro.

O treinamento ocorre em duas fases. Na fase *forward* (para frente) a saída da rede é definida. Em uma segunda fase, conhecida como *backward* (para trás), a saída atual é comparada com a saída desejada e atualizam-se os pesos das conexões (BULLINARIA, [ca. 2000]).

Nota-se que, como o erro é propagado da saída em direção à entrada, quanto maior a quantidade de camadas intermediárias, menos útil ele se torna no ajuste dos pesos.

A grande contribuição desse algoritmo foi solucionar o problema da definição do erro dos neurônios das camadas intermediárias, possibilitando com isso a atualização dos pesos intermediários de uma rede multicamadas.

A retropropagação do erro é bastante simples. A medida de erro a ser minimizada é a soma dos erros quadráticos:



$$\varepsilon(t) = \frac{1}{2} \sum_{i=1}^k (d_i(t) - y_i(t))^2$$

em que

$d_i(t)$  = i-ésima saída desejada no t-ésimo ciclo de treinamento; e

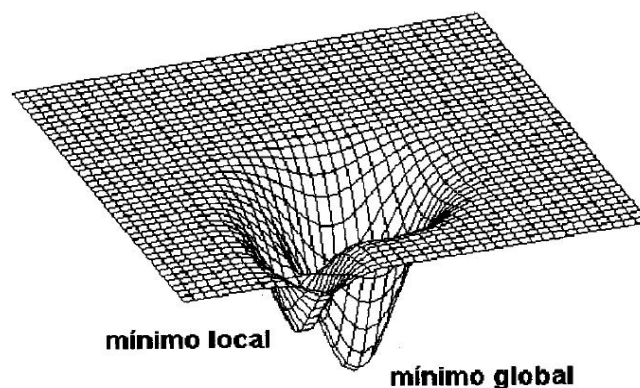
$y_i(t)$  = i-ésima saída observada no t-ésimo ciclo de treinamento.

Lembramos que o algoritmo da retropropagação do erro requer que as funções de ativação sejam contínuas e diferenciáveis, pois a alteração dos pesos é definida de acordo com o gradiente do erro em relação ao peso.

O treinamento de uma rede multicamadas por meio do algoritmo da retropropagação do erro pode ocorrer em duas periodicidades diferentes: *on-line* ou *batch*. Quando os pesos são atualizados após a apresentação de cada padrão de treinamento, denomina-se treinamento *on-line*. Quando os pesos são atualizados após todos os padrões serem apresentados (ao fim de um ciclo), denomina-se treinamento *batch*.

**Problemas no treinamento.** O treinamento não implica garantia de implementação da função desejada. Há casos de convergência para mínimos locais (Figura 9), de demora excessiva e de *overfitting* (a rede fica especializada para os dados à qual foi treinada e tem uma generalização ruim para dados desconhecidos).

Outro problema encontrado é a convergência para uma solução de mínimo local, que, embora seja uma solução estável, não é mínima.



Fonte: disponível em: <[www.inf.ufrgs.br/procpa/disc/cmp135/trabs/rafael/T2/RNA1.ppt](http://www.inf.ufrgs.br/procpa/disc/cmp135/trabs/rafael/T2/RNA1.ppt)>.

Figura 9 – Superfície de erro, evidenciando um mínimo local (resultado a ser evitado) e um mínimo global (resultado desejado).

As alternativas para reduzir a ocorrência de mínimos locais são (BRAGA *et al.*, 1998):

- Utilizar uma taxa de aprendizado decrescente.
- Acrescentar neurônios intermediários.
- Acrescentar o termo *momentum*.
- Adicionar ruído aos dados.

Das alternativas apresentadas, a mais utilizada é a adição do termo *momentum*, resultando na seguinte regra para atualização dos pesos (BRAGA *et al.*, 1998):

$$w_{ji}(t+1) = w_{ji}(t) - \eta \frac{\partial \varepsilon(t)}{\partial w_{ij}} + \mu(w_{ij}(t) - w_{ij}(t-1))$$

Quanto ao *overfitting*, que é a perda de capacidade de generalização, as alternativas apresentadas por Braga *et al.* (1998) são:

- Encerrar o treinamento mais cedo.
- Aplicar a técnica de poda (*pruning*).

**Redes RBF.** Outros tipos de redes que têm apresentado excelentes resultados são as redes RBF, as redes construtivas e as de processamento temporal. O interesse pelas redes RBF vem crescendo em aplicação.

As redes RBF possuem camadas ocultas e de saída com regras completamente diferentes, assim como os pesos. Portanto, fica claro que a técnica de aprendizado para as redes RBF é diferente das redes multicamadas.

Nas redes multicamadas, a função de ativação, aplicada a cada nodo, utiliza como argumento o produto escalar do vetor de entrada ( $x$ ) e do vetor de peso ( $w$ ) associado a esse nodo. No caso das redes RBF, a função de ativação tem como argumento a distância entre seus vetores de entrada e de peso.

Comparando as redes multicamadas com as redes RBF, nota-se que na primeira os nodos intermediários utilizam funções de base sigmoideal. No segundo tipo de rede, os nodos intermediários utilizam funções de base radial (BULLINARIA, [ca. 2000]).

As redes RBF possuem uma arquitetura típica, sendo uma camada de entrada, uma camada de saída e uma única camada intermediária. Como mencionado anteriormente, cada camada possui um papel específico.

A camada intermediária, que possui os nodos com funções de ativação de base radial, agrupa os dados em *clusters*. Nesta camada, o problema passa de não-linearmente separável para linearmente separável. A camada de saída classifica os padrões recebidos da camada anterior.

As funções de base radial são  $\phi_j = (v, \mu_j, \sigma_j)$ , em que  $v$  é a distância euclidiana  $\|x - \mu\|$ ,  $\mu$  é o centro e  $\sigma$  é a largura da função base radial.

A resposta  $y$  de um nodo  $j$  é definida como sendo:

$$y_j = \sum_{i=1}^k w_{ji} \phi(\|x - \mu_i\|) + w_{j0}$$

Recomenda-se utilizar um número de funções radiais menor que o número total de padrões de treinamento.

As funções de base radial comumente utilizadas são: a função gaussiana, a função multiquadrática, ou a função *thin-plate-spline* (HAYKIN, 2001).

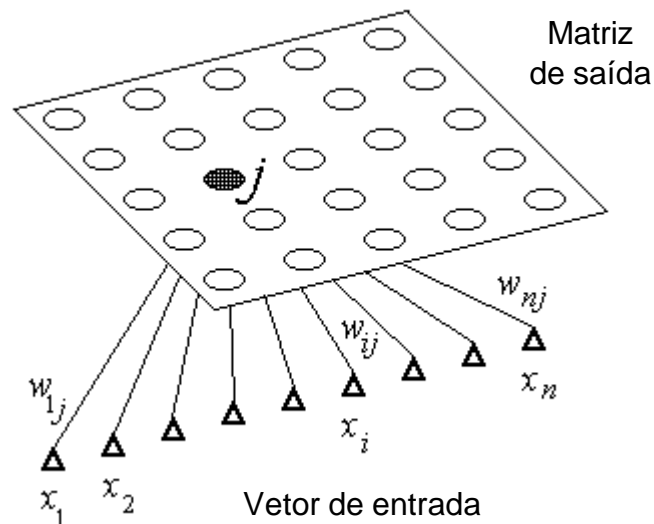
O treinamento é dividido em dois estágios. No primeiro, o número de funções radiais e seus parâmetros são determinados por métodos não-supervisionados.

Nesta primeira fase, diferentes abordagens podem ser adotadas, como:

- selecionar aleatoriamente os centros das funções radiais; e
- utilizar técnicas de clusterização. Geralmente, se aplica o algoritmo *k-means*.

**Redes Auto-Organizáveis (SOM).** As redes auto-organizáveis (do inglês *self-organising* – redes SOM) são conhecidas também como redes SOFM (*self-organising features maps*) ou redes Kohonen (Figura 10).

Nas redes SOM, os neurônios estão espacialmente organizados em forma de reticulado. Assim, neurônios topologicamente próximos tendem a responder a padrões ou estímulos semelhantes. Isto só é possível devido ao uso de uma função tipo “chapéu mexicano”.



Fonte: disponível em: <<http://www.cies.staffs.ac.uk/somview/som-vect.gif>>).

Figura 10 – Arquitetura de uma rede SOM.

Essa função divide o espaço em três regiões:

- excitatória: quando os neurônios vizinhos estão dentro de um raio  $R_1$ ; e
- inibitória: quando os vizinhos estão fora do raio  $R_1$ , porém dentro do raio  $R_2$  sendo  $R_2 > R_1$ .

- Levemente excitatória: neste caso, os neurônios vizinhos estão em uma área superior ao raio  $R_2$ , porém dentro de um raio  $R_3$ , sendo  $R_3 > R_2 > R_1$ .

Quando um padrão de entrada  $[p]$  é apresentado e uma unidade é muito parecida com  $[p]$ , a semelhança desta unidade e de seus vizinhos é aumentada durante o treinamento. Em outras palavras, quando um nodo vence a competição (produz a maior saída para uma dada entrada), ele e seus vizinhos têm seus pesos ajustados.

Nota-se que nas redes SOM os nodos são localmente interconectados, e o processo de adaptação é restrito ao nodo vencedor e seus vizinhos.

Por se organizarem topologicamente, os padrões detectados por um neurônio estão relacionados com as coordenadas da posição deste neurônio dentro do reticulado.

Logo, as localizações espaciais guardam características dos padrões classificados. Assim, o estado de ativação de um nodo é determinado pela distância entre o seu peso e o vetor de entrada.

$$y_j = \sum_{i=1}^n \|x_i - w_{ji}\|$$

em que  $y_j$  é o valor de ativação do neurônio  $j$ .

O treinamento das redes SOM é competitivo. A cada novo padrão de treinamento apresentado, os neurônios competem pelo maior valor de saída. Aquele neurônio vencedor tem seu peso atualizado junto com seus vizinhos. Durante o treinamento, a taxa de aprendizado e o raio de vizinhança são constantemente decrescidos.

O treinamento das redes SOM ocorre em duas fases: ordenação e convergência. Na fase de ordenação busca-se agrupar os neurônios do reticulado em *clusters*, de modo a refletir a distribuição dos padrões de entrada. Ao final dessa fase, a rede sabe quantos *clusters* identificar e onde eles se encontram no mapa.

Na segunda fase, onde a taxa de treinamento é baixa, assim como o raio de vizinhança, ocorre um refinamento do mapa, aprimorando o agrupamento previamente realizado.

Os principais componentes das redes SOM são:

- Inicialização: todos os pesos são inicializados com pequenos valores aleatórios.

- Competição: para cada padrão de entrada, os neurônios computam o valor de ativação, e aquele que apresentar o maior valor é considerado o vencedor.

- Cooperação: o espaço de vizinhança do neurônio vencedor é estabelecido.

- Adaptação: os neurônios excitados têm os valores de ativação aumentados.

Os passos para se obter uma rede SOM ajustada são os seguintes, segundo Bullinaria [ca. 2000]:

1. Inicialização: atribuir valores aleatórios aos pesos.
2. Amostragem: escolher um vetor  $x$  do espaço de entrada.
3. Cálculo: encontrar o neurônio vencedor de menor distância euclidiana entre o vetor de entrada e o vetor de pesos.
4. Atualização:

$$\Delta w_{ji} = \eta(t) T_{j,l(x)}(t) (x_i - w_{ji})$$

5. Seqüência: retornar ao passo dois até o mapa não sofrer alterações.

**As tendências.** Uma promissora área de pesquisa, segundo Braga *et al.* (1998), é a combinação de técnicas para solução de problemas, que quando tratados isoladamente por uma das técnicas não apresentam solução satisfatória.

A idéia é justamente combinar técnicas com a finalidade de obter uma solução mais eficiente.

Braga *et al.* (1998) relataram que um modelo de computação verdadeiramente inteligente tem maiores chances de ser desenvolvido se conseguir aglutinar potenciais de diferentes técnicas da inteligência artificial.

Estuda-se atualmente a integração de RNAs com:

- Lógica Fuzzy.
- Algoritmos genéticos.
- Raciocínio baseado em dados.
- Sistemas especialistas.
- Agentes Inteligentes, e muitos outros.

### 3. PRÉ-PROCESSAMENTO

A mineração de dados é a etapa mais popularizada do processo de descoberta de conhecimento. No entanto, todo processo de mineração deve ser iniciado por uma etapa de preparação dos dados. A qualidade do conhecimento a ser gerado depende da informação disponível.

Pode-se considerar o estudo de volumetria como um processo de descoberta de conhecimento. Os processos de descoberta de conhecimento são tecnicamente denominados de KDD (*Knowledge Discovery in Databases*).

Em essência, o KDD é sempre orientado a um objetivo. Desta forma, diferentes ferramentas podem ser utilizadas, independentemente ou combinadas, para se atingir tal objetivo. Nesta dissertação, o processo de descoberta de conhecimento focado é o de redes neurais.

**Pré-processamento.** Consiste em manipular as informações para adequá-las às ferramentas de descoberta de conhecimento, neste caso redes neurais artificiais. Assim como todas as outras, esta possui algumas exigências quanto à apresentação dos dados, sendo, portanto, indispensável a realização de um pré-processamento.

O pré-processamento é uma tarefa singular. Se feito com exagero, o analista perderá qualidade no conhecimento gerado. Por outro lado, caso esta etapa seja feita com desleixo, o analista terá dificuldade em extrair algum tipo de conhecimento. Cada conjunto de dados necessitará de um pré-

processamento específico, cabendo ao analista determiná-lo. Em Goldschmidt e Passos (2005) e Hair *et al.* (2005), podem-se encontrar diversos procedimentos, dos quais os mais importantes para redes neurais artificiais serão abordados na seqüência.

**Redução de atributos (variáveis).** A redução de atributos pode ser independente ou dependente do modelo. Na primeira situação, a redução se dá antes da aplicação da técnica de KDD. Na segunda, analisam-se diferentes opções de redução mediante o comportamento dos dados durante a aplicação da técnica de KDD.

Nota-se que esse último é, sem dúvida, muito mais trabalhoso que o primeiro, e geralmente é utilizado quando não se possui um conhecimento *a priori* do universo estudado.

Tecnicamente pode-se utilizar tanto a redução de atributos direta quanto a indireta. A direta parte do conhecimento do analista sobre quais os principais atributos a serem utilizados no processo. Já a indireta se dá por meio de ferramentas estatísticas para se avaliar o grau de importância do atributo, ou utiliza-se uma ferramenta para realizar a redução.

**Redução de casos (linhas).** Consiste na escolha de um critério direto que divida o conjunto de casos em aptos e não-aptos. A amostragem pode ser usada para escolher quais os casos serão considerados aptos. Para tanto, são sorteadas da base de dados as informações que serão utilizadas durante o processo de descoberta de conhecimento.

Outra opção encontrada na literatura para reduzir casos é a agregação de informação, que nada mais é do que a união de um ou mais casos, sem perder, no entanto, suas características intrínsecas.

Alguns autores sugerem a redução de valores. Esta redução nada mais é do que a generalização da informação, ou a perda de resolução, que leva à extração de conhecimento mais rápida. Porém, perde-se um pouco de profundidade na análise. Um exemplo clássico de redução de valores é substituir as ruas pelos respectivos bairros.



**Limpeza.** É uma etapa importante para aumentar a qualidade do banco de dados. Busca-se, então, eliminar registros ausentes, incompletos ou distorcidos. Esta tarefa se dá usualmente por meio de filtros ou de rotinas de consistência. Deve-se lembrar que os dados não podem ser excluídos de um banco de dados, e sim descartados do processo de KDD.

Um dado ausente, geralmente, compromete toda a linha de informação. A melhor alternativa é excluir todo caso que apresente um registro ausente. No entanto, podemos encontrar na literatura algumas alternativas à exclusão, entre os quais se destacam a substituição pelo valor médio do atributo e o preenchimento com valores históricos mais prováveis (BISHOP, 1995).

A presença de dados incompletos geralmente culmina no descarte da linha. Eventualmente, caso o registro se torne disponível, completa-se o banco de dados, buscando torná-lo o mais correto possível.

A alteração de valores deve ser evitada, pois esta leva a questionamentos quanto à manipulação. Deve-se preferir sempre o descarte de toda e qualquer irregularidade.

Toda inconformidade deve ser registrada, assim como a medida corretiva adotada.

**Transformação.** A principal aplicação deste pré-processamento é tornar um atributo adequado para aplicação da técnica de redes neurais. Pode-se verificar grande aplicabilidade na transformação de variáveis nominais em variáveis numéricas. Outra transformação muito interessante é a discretização. Neste caso, variáveis contínuas são transformadas em intervalos. O funcionamento das redes neurais artificiais é fundamentado em atributos numéricos, podendo ser tanto discretos quanto contínuos. Daí a importância deste pré-processamento.

**Enriquecimento.** É uma tarefa de pesquisa a fontes externas, de forma a completar o banco existente com alguma informação relevante. Incluir novas informações pode, em alguns casos, aumentar a resolução ou o poder preditivo da rede. No entanto agregar informações de forma desorganizada, ou redundante, pode atuar negativamente no resultado, levando à maior lentidão da rede, ou até mesmo prejudicando a tarefa de predição.

**Normalização.** Sem dúvida é o pré-processamento principal. Segundo Haykin (2001), para evitar a saturação dos neurônios, os dados devem estar normalizados. A saturação ocorre quando um atributo de magnitude muito alta polariza a rede em sua direção. Para evitar isso, é importante que ocorra a normalização dos dados, permitindo que a saída pondere todos os *inputs* de forma equivalente. Dois intervalos são tratáveis em redes neurais: de 0 a 1, ou de -1 a 1. O primeiro intervalo é o mais popular.

Vale lembrar que dentro de cada um dos neurônios existe uma função de ativação. Esta função recebe o somatório das entradas ponderadas pelos respectivos pesos, e aplica a função. Dentre as diversas funções existentes para se aplicar o algoritmo da retropropagação do erro (*backpropagation*), essas devem ser diferenciáveis. As mais comuns são: exponencial, logística e tangente hiperbólica. As duas primeiras possuem contradomínio de [0,1]. Já a tangente hiperbólica possui contradomínio de [-1,1].

A saturação ocorre quando o neurônio recebe sempre valores muito altos, o que leva à função de ativação a trabalhar em sua região assintótica, causando pouca diferenciação dos valores de entrada durante o processamento.

Diversas formas de normalização são encontradas na literatura, como: a linear, pelo desvio-padrão, pela soma, pelo valor máximo e pela escala decimal.

Neste estudo não será abordada a normalização por desvio-padrão. Este tipo de normalização reduz a amplitude dos dados para próximo de 0 a 1, porém, na realidade, sua variância será de 1 e o valor médio 0, mas os dados não ficarão restritos ao intervalo [0, 1].

**Normalização linear.** Consiste na interpolação dos valores não normalizados entre os limites superior e inferior estabelecidos pelo analista. A fórmula utilizada necessitará de informações de máximo e mínimo dos dados originais que se deseja normalizar.

$$V' = \frac{(V - \text{Min})(\text{LS} - \text{LI})}{(\text{Max} - \text{Min})} + \text{LI}$$

em que

V' é o valor normalizado;

LS e LI são os limites estabelecidos pelo analista, dentro dos quais os dados serão normalizados, por exemplo, 1 e 0, respectivamente;

Max e Min são informações sobre os dados não normalizados; e  
V é o valor a ser normalizado.

**Normalização pela soma.** Neste procedimento, dividi-se o valor a ser normalizado pela soma dos valores não-normalizados. Uma grande desvantagem deste método é o excesso de valores muito pequenos, que podem levar à saturação da rede.

**Normalização pelo valor máximo.** O valor máximo dividirá o valor não-normalizado. Sem dúvida, é um procedimento bastante simples e rápido. O valor máximo pode ser obtido pela análise dos dados, pela experiência do analista ou por dados históricos.

$$V' = \frac{V}{\text{Max}}$$

em que

V' = dado normalizado;

V = dado bruto; e

Max = valor máximo dos dados.

**Normalização pela escala decimal.** Consiste na alteração do intervalo de determinado atributo pela manipulação das casas decimais.

$$V' = \frac{V}{10^j}$$

em que

V' = dado normalizado;

V = dado bruto; e

j = número de zeros para reduzir o máximo valor ao intervalo de 0 a 1.

**Equalização.** Com certeza este é um termo muito pouco utilizado no setor florestal, porém muito importante para aplicação da técnica de redes

neurais artificiais. A equalização consiste em distribuir melhor os dados dentro de determinado intervalo, sem no entanto alterar sua distribuição ou suas características intrínsecas.

Geralmente, esse procedimento está acoplado à normalização dos dados. Caso se verifique, após a normalização, que os dados estão muito concentrados no intervalo de 0 a 0,5, é comum realizar novamente a normalização dos dados, induzindo o posicionamento entre 0,25 e 0,75. Note que a estrutura dos dados não se altera, porém os dados se deslocam por inteiro dentro do intervalo.

**Partição dos dados.** Consiste na divisão do conjunto de dados em dois ou mais, de forma que determinados conjuntos sejam utilizados no treinamento da rede e outros, na validação da rede treinada.

A partição é essencial para o bom treinamento de uma rede neural artificial. A eficiência de uma rede deve ser analisada em um conjunto de dados diferentes do treinamento.

A partição se dá geralmente de forma aleatória, em que determinado caso possui uma probabilidade “p” de ser alocado no conjunto de treinamento, e uma probabilidade “1 - p” de ser alocado no conjunto de validação.

A técnica de redes neurais não exige um conjunto de treinamento muito numeroso. A representatividade dos casos é mais importante que o número total de casos. Assim, a qualidade é fundamental, em vez da quantidade.

Vale ressaltar que alguns dos pré-processamentos exigem que, ao final da técnica de KDD, os valores sejam restaurados à escala original. O uso correto das técnicas de pré-processamento melhora muito as técnicas de KDD, permitindo que estas trabalhem de forma eficiente.

## ESTIMAÇÃO DO VOLUME DE ÁRVORES UTILIZANDO REDES NEURAIS ARTIFICIAIS

**Resumo:** Rede neural artificial consiste em um conjunto de unidades que contêm funções matemáticas, unidas por pesos. As redes são capazes de aprender, mediante modificação dos pesos sinápticos, e generalizar o aprendizado para outros arquivos desconhecidos. O projeto de redes neurais é composto por três etapas: pré-processamento, processamento e, por fim, um pós-processamento dos dados. Um dos problemas clássicos que podem ser abordados por redes é a aproximação de funções. Neste grupo pode-se incluir a estimação do volume de árvores. Foram utilizados quatro arquiteturas diferentes, cinco pré-processamentos e duas funções de ativação. As redes que se apresentaram estatisticamente iguais aos dados observados também foram analisadas quanto ao resíduo e à distribuição dos volumes e comparadas com a estimação de volume pelo modelo de Schumacher e Hall. As redes neurais formadas por neurônios cuja função de ativação era exponencial apresentaram estimativas estatisticamente iguais aos dados observados. As redes treinadas com os dados normalizados pelo método da interpolação linear e equalizados apresentaram melhor desempenho na estimação.

Palavras-chave: redes neurais; cubagem.

## ESTIMATING TREE VOLUME USING ARTIFICIAL NEURAL NETS

**Abstract:** Artificial neural network consists of a set of units containing mathematical functions, connected by weights. Such nets are capable of learning by means of synaptic weight modification, generalizing learning for other unknown archives. The neural network project is made up of three stages: data pre-processing, processing and post-processing. One of the classical problems approached by networks is function approximation. Within this group one can include tree volume estimate. Four different architectures, five pre-processings and two activation functions were used. The nets which were statistically similar to the data observed were also analyzed in relation to residue and volume and compared to volume estimate by the Schumacher and Hall model. The neural nets formed by neurons whose activation function was exponential, presented estimates statistically similar to the observed data. The nets trained with the data normalized by the linear interpolation method and equalized presented better estimate performance. Keywords: neural nets; tree scaling.

Key words: neural networks; scaling.

## 1. Introdução

Redes neurais artificiais (RNAs) são sistemas massivos e paralelos, compostos por unidades de processamento simples que computam determinadas funções matemáticas (BRAGA *et al.*, 1998). Utilizando um conjunto de exemplos apresentado, as RNAs são capazes de generalizar o conhecimento assimilado para um conjunto de dados desconhecidos. Elas têm ainda a capacidade de extrair características não-explícitas de um conjunto de informações que lhes são fornecidas como exemplos (KOVÁCS, 1996).

Um projeto de redes neurais consiste em pré-processamento, processamento e, por fim, um pós-processamento dos dados.

Os problemas tratáveis por meio de redes neurais podem ser divididos em três tipos de problemas principais: aproximação de função, classificador de padrões e agrupamento de dados (HAYKIN, 2001). Em Ciência Florestal, diversos problemas podem ser potencialmente tratados por RNAs, dentre eles a modelagem do volume de árvores.

Um problema importante ao conduzir inventários florestais é a definição de equações volumétricas apropriadas para cada estrato, sendo este geralmente determinado pelo material genético, pelo espaçamento inicial, pelo regime de corte e pela idade, e, então, geradas equações específicas, atualizadas anualmente. Com isso, empresas de grande porte gastam quantias consideráveis de recursos humanos e financeiros com o abate de árvores ao longo do tempo (LEITE e DE ANDRADE, 2002).

Apesar da contínua atualização de equações volumétricas, em muitas ocasiões não é possível gerar equações para determinados compartimentos. Por exemplo, no caso de plantios clonais e de pesquisa (testes clonais), sempre há estratos onde não é possível abater árvores para ajuste de modelos volumétricos, sendo aplicadas equações de outros estratos semelhantes (CAMPOS e LEITE, 2006).

O fato é que ao longo dos anos as empresas acumulam grande quantidade de dados de cubagem, incluindo árvores com diferentes características e variações de forma (LEITE *et al.*, 1995). Esses dados podem ser utilizados para implementar um projeto de redes neurais cujo objetivo final é

a estimativa do volume de árvores sem o uso de equações volumétricas e de dados específicos de cubagem.

O objetivo deste estudo foi construir uma rede neural que estime com eficiência o volume de árvores.

## 2. Material e Métodos

Foram coletados dados de cubagem de eucalipto (*Eucalyptus* spp.) em quatro empresas diferentes, sendo uma no Estado da Bahia, outras duas no Estado de Minas Gerais e a quarta no Estado de São Paulo. Dados de cubagem em teca (*Tectona grandis* L.f.) no Estado do Mato Grosso também foram utilizados.

Tabela 1 – Análise descritiva dos dados de cada local

Local	Estatística	dap	alt	volume
SP	Média	9,7155	9,0860	0,0380
	Desvio-padrão	2,5106	2,0043	0,0223
	Mínimo	3,0000	3,4000	0,0015
	Máximo	15,6000	13,4000	0,1094
BA	Média	18,0521	24,0611	0,3413
	Desvio-padrão	6,4886	5,2191	0,2495
	Mínimo	7,0000	11,5000	0,0231
	Máximo	31,0000	35,0000	1,0093
MG1	Média	24,6673	27,2127	0,5664
	Desvio-padrão	5,5074	6,2084	0,3098
	Mínimo	16,5000	16,7000	0,1472
	Máximo	34,4000	36,3000	1,1093
MG2	Média	18,1527	27,3071	0,4126
	Desvio-padrão	6,2939	6,1370	0,3719
	Mínimo	8,6500	15,7800	0,0476
	Máximo	37,7500	47,6100	2,3597
MT	Média	9,8667	8,0633	0,0344
	Desvio-padrão	2,8660	1,9049	0,0195
	Mínimo	5,7000	5,1000	0,0108
	Máximo	15,9000	12,2000	0,0766



Foi definido como procedimento-padrão o ajuste do modelo de Schumacher e Hall (1933) para cada um desses locais, resultando em cinco equações diferentes.

Foram utilizadas redes ântero-alimentadas (*feedforward*), treinadas por meio do algoritmo da retropropagação do erro (*backpropagation*). Os dados de entrada para a rede foram: diâmetro a 1,30 m de altura (dap), altura total (ht), os cinco locais (representados por variáveis binárias) e as duas espécies (também representados pelas variáveis binárias) (Figura 1). Como exemplo, um padrão de entrada para uma árvore de eucalipto com dap normalizado de 0,2781 e altura normalizada de 0,3124 no local 5 seria representado pelo vetor (0,2781, 0,3124, 0, 0, 0, 0, 0, 1, 1, 0) = (dap, ht, L1, L2, L3, L4, L5, E1, E2).

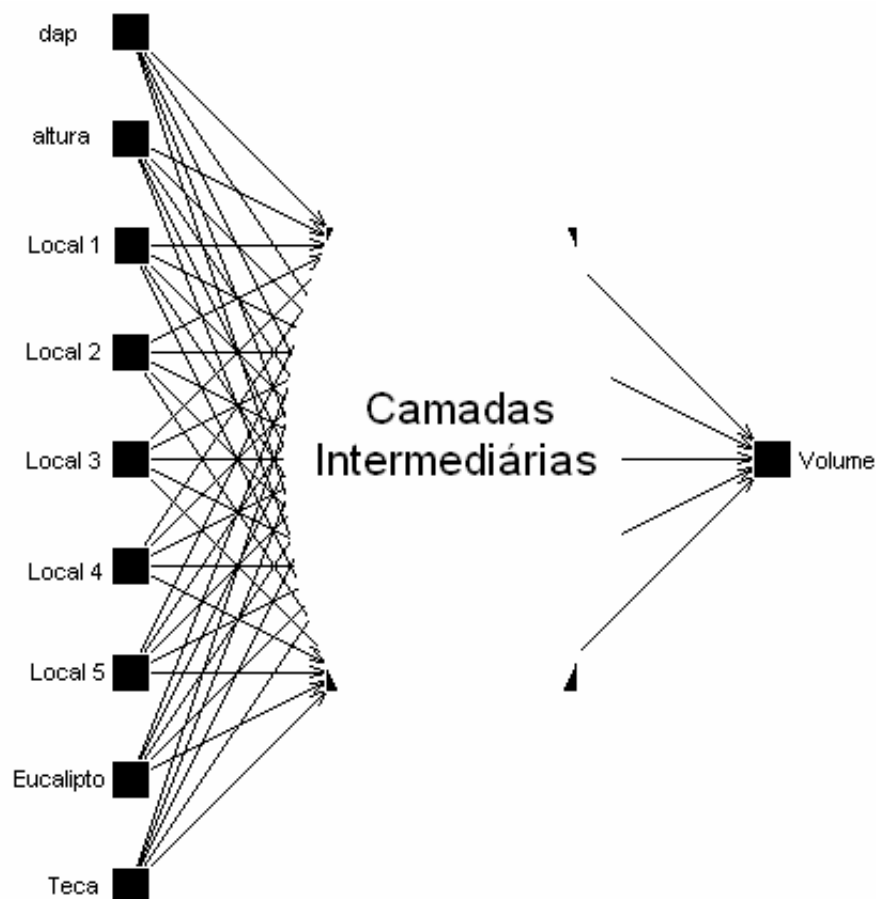


Figura 1 – Esquema com os neurônios de entrada e o neurônio de saída.

Como os dados provêm de cinco diferentes locais e empresas, optou-se por utilizar variáveis que discriminassem os locais. O mesmo ocorreu para espécie, pois o crescimento e o comportamento da teca são diferentes daqueles observados para o eucalipto.

A etapa de pré-processamento foi realizada, utilizando diferentes procedimentos encontrados na literatura (GOLDSCHMIDT e PASSOS, 2005), sendo eles:

- 1) Valor observado, dividido pelo valor máximo da variável.

$$\text{valor transformado} = \frac{x}{x_{\text{máximo}}}$$

Este é um método de normalização no qual o valor máximo pode ser obtido por experiência do usuário, por análise do banco de dados ou série histórica.

- 2) Valor observado dividido por um múltiplo de 10, de forma a alterar as casas decimais dos números, para que se incluam no intervalo de 0 a 1.

$$\text{valor transformado} = \frac{x}{10^n}$$

Esse também é apenas um método de normalização, não se preocupando com a equalização dos dados.

- 3) Interpolação linear inversa entre 0 e 1.

$$\text{valor transformado} = 1 + \left( \frac{-x + \text{Mínimo}}{\text{Máximo}} \right)$$

A interpolação linear pode ser adaptada para realizar tanto a normalização quanto a equalização, como nos pré-processamentos 4 e 5.

4) Interpolação linear inversa entre 0 e 0,8.

$$\text{valor transformado} = 0,8 + \left( \frac{-x + \text{Mínimo}}{\text{Máximo}} \right)$$

5) Interpolação linear entre 0,1 e 0,8.

$$\text{valor transformado} = 0,1 + \left( 0,8 \frac{x - \text{Mínimo}}{\text{Máximo} - \text{Mínimo}} \right)$$

Três etapas são necessárias para se obter uma rede pronta para ser usada: construção da rede, treinamento e validação. Neste estudo foram avaliadas quatro arquiteturas diferentes.

Tabela 2 – Arquiteturas avaliadas para o problema da estimação do volume de árvores

Arquitetura	1	2	3	4
Função de ativação	Logística	Exponencial	Logística	Exponencial
Nodos ocultos 1	5	5	10	10
Nodos ocultos 2	5	5	8	8

O algoritmo de aprendizado usado para o treinamento das redes foi o da retropropagação do erro (*backpropagation*), e os parâmetros de treinamento foram taxa de aprendizagem ( $\zeta$ ) de 0,2 e termo *momentum* ( $i$ ) de 0,9. O *software* utilizado neste trabalho foi o JavaSNNS (ZELL *et al.*, 1998) (disponível em: <<http://www-ra.informatik.uni-tuebingen.de/SNNS/>>), e o treinamento da rede foi encerrado quando o gráfico do erro tendia à estabilização.

No pós-processamento foi utilizada uma transformação inversa ao pré-processamento, acompanhada de uma regra lógica: “se  $X < 0$ , então  $X = 0$ , caso contrário  $X = X$ ”. Tal regra foi importante para assegurar que não ocorressem valores inferiores a zero (realismo biológico).

Os resultados das redes foram comparados com os valores observados, e com os valores estimados pela equação de Schumacher e Hall, por meio do teste t, de gráficos de dispersão XY e de distribuição dos dados.

Aquelas redes que se apresentaram estatisticamente iguais foram analisadas graficamente e as demais foram descartadas das análises subseqüentes.

### 3. Resultados e Discussão

O pré-processamento 2, divisão por múltiplo de dez, não apresentou ajustes confiáveis das redes, sendo desnecessárias as análises subseqüentes.

Para melhor identificação, as redes foram numeradas de acordo com a seqüência de trabalho, não significando classificação ou preferência (Tabela 3).

Tabela 3 – Identificação das redes de acordo com o pré-processamento e com a arquitetura de rede selecionada

Pré-Processamento	Arquitetura			
	1	2	3	4
1	9	28	29	30
3	13	15	16	17
4	26	18	19	20
5	25	21	22	23

Os volumes estimados pelas redes 13, 15, 18, 20 e 26 foram estatisticamente iguais, pelo teste t, aos volumes observados correspondentes. Nas redes 13 e 15 foi aplicado o pré-processamento 3 e nas redes 18, 20 e 26, o pré-processamento 4.

Quanto à arquitetura, as redes 13 e 26 apresentaram duas camadas intermediárias com cinco neurônios cada e função de ativação logística. As redes 15 e 18 foram formadas por duas camadas intermediárias com cinco neurônios cada e função de ativação exponencial. A rede 20 possuía duas camadas intermediárias com 10 e 8 neurônios e função de ativação exponencial.

Apenas as redes 13, 15, 18, 20 e 26 foram estatisticamente iguais aos dados observados diante do teste  $t$ , para 5% de probabilidade. As redes que apresentaram os valores estimados estatisticamente iguais aos observados foram analisadas quanto à dispersão dos dados observados *versus* os valores estimados (Figura 2). Estas mesmas redes também foram analisadas quanto à distribuição dos volumes estimados diante da distribuição real dos dados (Figura 3).

As redes 13 e 18 apresentaram subestimação dos volumes para classes pequenas. A rede 20 apresentou tendência de subestimação geral. A dispersão da rede 26 apresentou característica sigmoideal, superestimando classes baixas e subestimando classes altas.

É fundamental que determinado procedimento mantenha a mesma distribuição da resposta objetivo. Para visualizar esta propriedade, plotam-se as distribuições de freqüência dos volumes.

As redes 13 e 26 apresentaram distribuição de diâmetros bastante alterada, comparada com a dos dados originais.

Os volumes obtidos pela metodologia tradicional foram estatisticamente iguais aos volumes observados ( $p > 0,05$ ). Quando se comparou a modelagem tradicional com a modelagem por meio de redes neurais, apenas a rede 15 (Figura 4) apresentou resultados estatisticamente iguais àqueles obtidos com equações ajustadas a partir do modelo de Schumacher e Hall (1933).

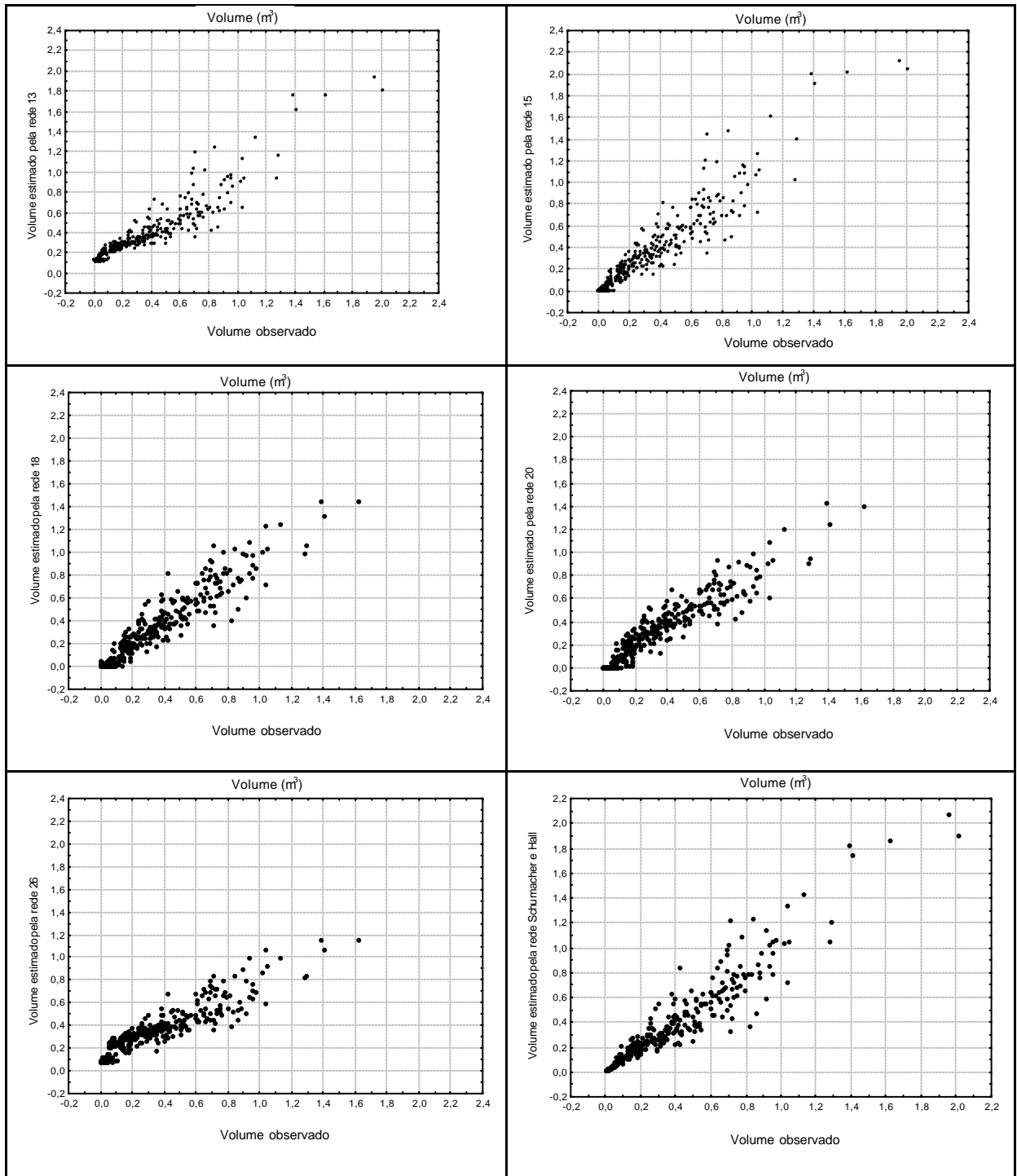


Figura 2 – Valores observados e estimados das diferentes redes em que os resultados foram estatisticamente iguais aos dados observados.

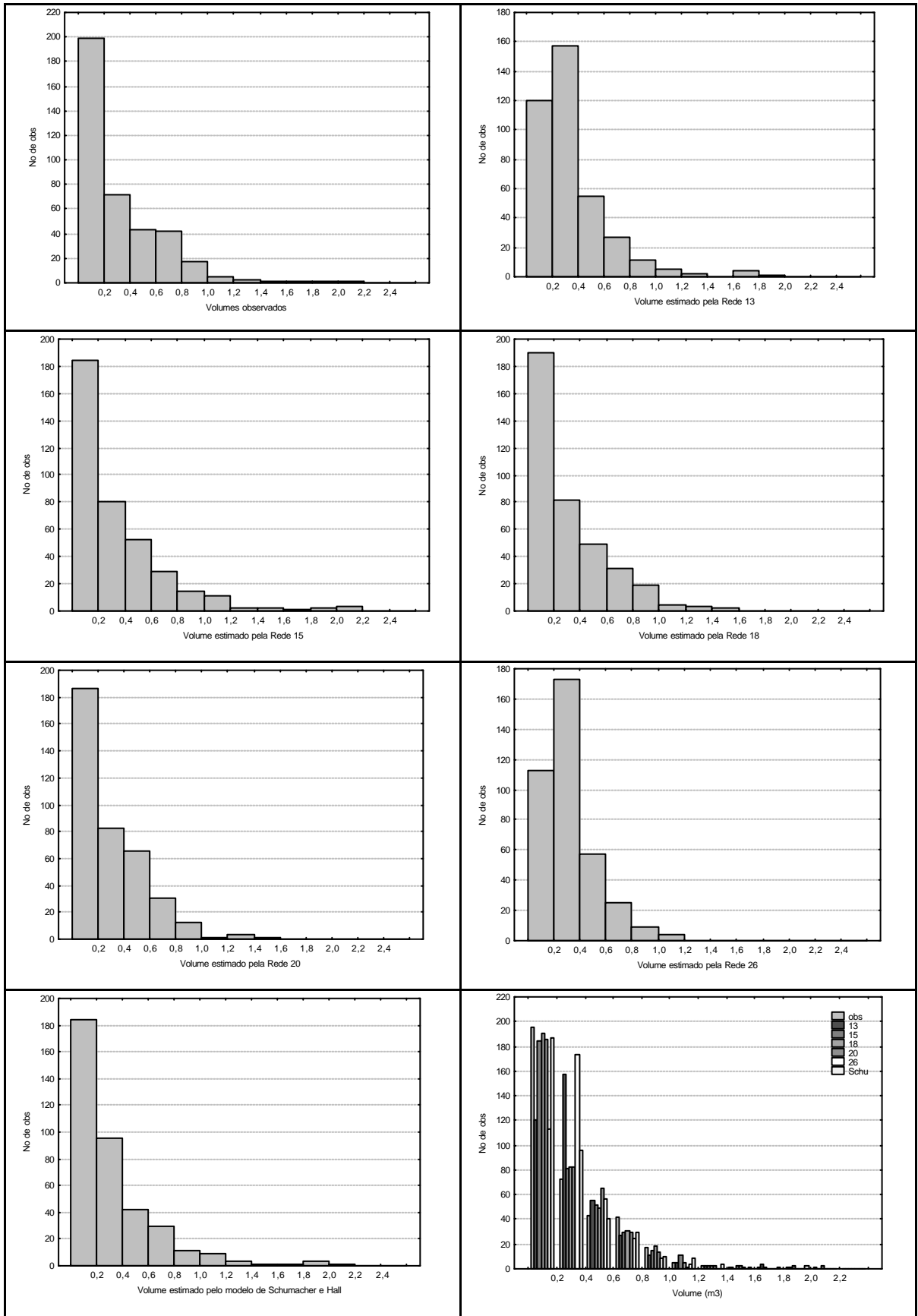


Figura 3 – Distribuição de freqüência dos volumes observados e dos volumes estimados pelas redes.

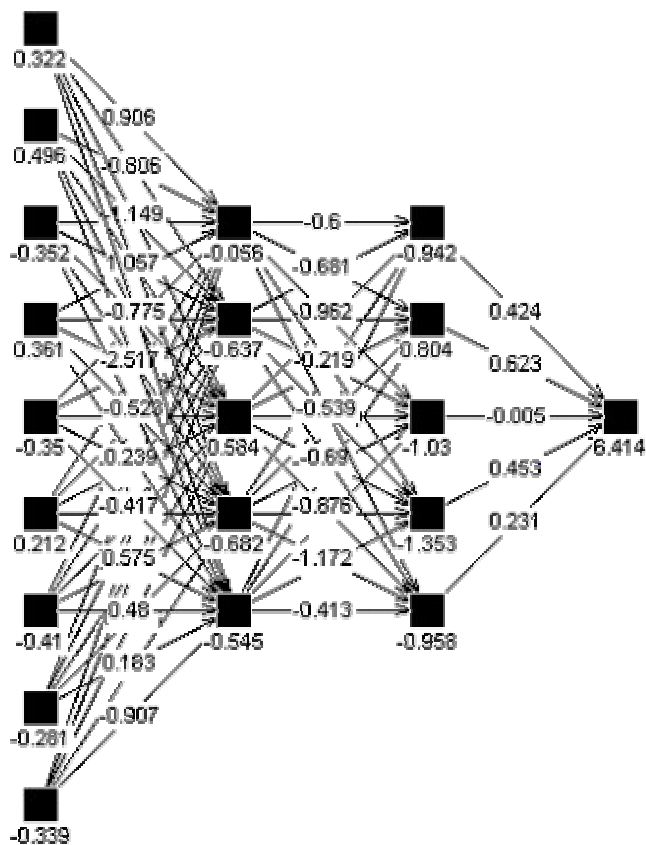


Figura 4 – Esquema da rede 15, em que os números sobre os arcos são os pesos e os números sob os quadrados são as bias.

#### 4. Conclusões

A interpolação linear como método de normalização fornece melhores resultados. A equalização dos dados melhora a capacidade de predição da rede neural.

As redes que apresentaram os melhores resultados possuíam a função exponencial como função de ativação. O número de neurônios na camada oculta não resultou em melhorias.

A rede apresentada foi capaz de substituir pelo menos dez equações, uma vez que a equação deve ser específica para cada local e espécie.

A modelagem por redes neurais é perfeitamente viável, devendo ser lembrado que apenas uma rede é capaz de realizar a predição de volume de árvores para os cinco locais e duas espécies.



## 5. Referências Bibliográficas

BRAGA, A. de P.; CARVALHO, A. P. de L. F. de; LUDEMIR, T. B. **Fundamentos de redes neurais artificiais**. Rio de Janeiro: DCC/I, COPPE/Sistemas, NCE/UFRJ, 1998. 246 p.

CAMPOS, J. C. C.; LEITE, H. G. **Mensuração florestal: perguntas e respostas**. 2. ed. rev. e ampl. Viçosa: UFV, 2006. 470 p.

GOLDSCHMIDT, R.; PASSOS, E. **Data mining: um guia prático**. Rio de Janeiro: Elsevier, 2005. 261 p.

HAYKIN, S. **Redes neurais: princípios e prática**. 2. ed. Porto Alegre: Bookman, 2001. 900 p.

KOVÁCS, Z. L. **Redes neurais artificiais: fundamentos e aplicações**. 2. ed. Ver. e amp. São Paulo: Colledium cognitio, 1996. 174 p.

LEITE, H. G.; ANDRADE, V. C. L. de. Um método para condução de inventários florestais sem o uso de equações volumétricas. **Rev. Árvore**, v. 26, n. 3, p.321-328, 2002.

LEITE, H. G.; GUIMARÃES, D. P.; CAMPOS, J. C. C. Descrição e emprego de um modelo para estimar múltiplos volumes de árvores. **Rev. Árvore**, v. 19, n. 1, p. 1-21, 1995.

SCHUMACHER, F.X.; HALL, F.S. Logarithmic expression of timber-tree volume. **Journal of Agricultural Research**, v. 47, n. 9, p. 719-734, 1933.

ZELL, A.; MAMIER, G.; MACHE, M. V. N.; HÜBNER, R.; DÖRIN, S.; HERMANN, K-U; SOYEZ, T.; SCHMALZL, M.; SOMMER, T.; HATZIGEORGIOU, A.; POSSELT, D.; SCHREINER, T.; KETT, B.; CLEMENTE, G.; WIELAND, J.; GATTER, J. **SNNS – Stuttgart Neural Network Simulator v.4.2**, User Manual, University of Stuttgart/ University of Tübingen, 1998. 338 p.

## DETERMINAÇÃO DA ARQUITETURA E DO PRÉ-PROCESSAMENTO ÓTIMOS PARA A ESTIMAÇÃO DE VOLUME DE ÁRVORES POR MEIO DE REDES NEURAIS ARTIFICIAIS

**Resumo:** As redes neurais supervisionadas são compostas por unidades de processamento organizadas de forma paralela, onde cada uma dessas computa determinadas funções matemáticas. As unidades são ligadas por pesos sinápticos que ponderam as entradas, buscando ajustá-los a um padrão de saída previamente estabelecido e, portanto, minimizando o erro para um ponto de mínimo global. Os dados de cubagem de cinco empresas diferentes foram reunidos em uma planilha e, de forma aleatória, divididos em conjunto de treinamento e conjunto de validação. Foram utilizados três pré-processamentos e três arquiteturas diferentes, resultando em nove redes testadas. A avaliação foi feita por meio de gráficos de resíduos e teste estatístico, comparando valores estimados com valores observados. As redes que apresentaram estimativas estatisticamente iguais e gráfico de resíduos livre de bias foram analisadas quanto à distribuição dos resíduos por rede e por local. As estimativas das redes 46, 48 e 50 apresentaram-se estatisticamente iguais, pelo teste t, aos valores observados e resíduos livres de bias. Com base nos resultados, os dados devem ser pré-processados e a rede deve possuir duas camadas intermediárias, sugerindo-se dez e oito neurônios, respectivamente.

Palavras-chave: redes neurais; pré-processamento; arquitetura.

## **DETERMINING OPTIMUM ARCHITECTURE AND PRE-PROCESSING FOR ESTIMATING TREE VOLUME BY ARTIFICIAL NEURAL NETWORKS**

**Abstract:** Supervised neural networks are composed of processing units organized in parallel, with each computing specific mathematical functions. These units are connected by synaptic weights. The weights are adjusted thus minimizing error to a global minimum point. Tree scaling data of five different enterprises were combined and randomly grouped into training and validation set. Three different pre-processing and three architectures were used, resulting in nine tested nets. Evaluation was performed by means of residue graphics and statistical test, by comparing the estimated values with the observed values. The nets that presented statistically similar estimates and unbiased residue graphic were analyzed in relation to residue distribution per net and per location. The estimates of nets 46, 48 and 50 were statistically similar by the t test to the values observed and unbiased residues. Based on the results obtained, data must be processed and the net must have two intermediary layers, with ten and eight neurons being suggested, respectively.

**Keywords:** neural nets; pre-processing; tree scaling.

## 1. Introdução

Redes neurais artificiais (RNAs) são sistemas massivos e paralelos, compostos por unidades de processamento simples que computam determinadas funções matemáticas (BRAGA *et al.*, 1998). Utilizando um conjunto de exemplos conhecidos, as RNAs são capazes de generalizar o conhecimento assimilado para um conjunto de dados desconhecidos. Elas têm ainda a capacidade de extrair características não-explicítas de um conjunto de informações que lhes são fornecidas como exemplos (KOVÁCS, 1996).

Qualquer trabalho com redes neurais caracteriza-se por possuir muitos parâmetros, que em sua maioria são definidos de forma empírica. No que se refere à construção da rede, os principais pontos que cabem ao usuário definir são: o número de camadas intermediárias, o número de neurônios e a função de ativação dos neurônios (KAVZOGLU e MATHER, 2003).

Sendo uma rede do tipo *feedforward*, o principal algoritmo de treinamento é o da retropropagação do erro mais o termo *momentum*. Uma rede *feedforward* possui as ligações entre os neurônios ocorrendo sempre no sentido entrada-saída. Associado a cada uma dessas ligações existe um valor, denominado peso. Os principais parâmetros que o usuário deve definir são: a taxa de aprendizagem e o valor do termo *momentum*. O algoritmo da retropropagação do erro utiliza a direção contrária à do vetor gradiente para indicar o caminho a ser seguido em busca do conjunto de pesos que resulta no erro mínimo. A taxa de aprendizado é o tamanho do passo que o algoritmo dará na direção escolhida. O termo *momentum* é útil para evitar que o algoritmo fique preso à região do espaço de resposta conhecida como mínimo local (HAYKIN, 2001).

O algoritmo da retropropagação do erro é um algoritmo de aprendizado supervisionado, o que implica que a saída da rede, durante o treinamento, é comparada à saída desejada e previamente conhecida. Esta comparação é que orienta o caminhamento de aprendizagem da rede, que nada mais é do que o ajuste dos pesos.

É desejável que os dados sejam apresentados à rede já normalizados. Caso contrário, pode ocorrer uma polarização desta, priorizando os neurônios e as sinapses que recebem os maiores valores, em detrimento das outras (BISHOP, 1995).

A normalização e a equalização podem estar incluídas no pré-processamento. Como pode ser constatado em Goldschmidt e Passos (2005), existem diversas formas de normalização dos dados. A mais usada é a interpolação linear, pois ela também pode realizar a equalização dos dados.

A equalização dos dados é outro procedimento que pode ser aplicado aos dados que serão apresentados à rede. Ela é especialmente interessante quando os dados apresentarem grande concentração em classes pequenas, ou em classes elevadas, resultando em distribuição desequilibrada dos dados.

Ao fim do treinamento, os resultados da rede devem ser redimensionados à escala original, por meio da operação inversa do pré-processamento. Em alguns casos, os dados podem sofrer algum tratamento, antes de serem considerados como resultado. Desta forma, um projeto de rede é composto por três etapas: pré-processamento, processamento e pós-processamento.

Neste trabalho, foram avaliados três diferentes arquiteturas e três diferentes pré-processamentos para o problema da aproximação de função de estimação de volume de árvores.

## 2. Material e Métodos

Os dados foram obtidos de cinco empresas florestais, sendo uma localizada no Estado da Bahia, uma no Estado de São Paulo e duas em Minas Gerais. Nestas quatro empresas, os dados foram obtidos em plantios de eucalipto (*Eucalyptus* spp.). Uma quinta empresa, do Estado do Mato Grosso, também foi incluída no trabalho, neste caso com dados de teca (*Tectona grandis* L. f.). Para cada uma dessas empresas, denominadas por local 1 a local 5, foram obtidos dados de diâmetro a 1,30 de altura (*dap*), altura total (*ht*) e o volume por árvore cubada, sendo 164, 55, 207, 30, 532 árvores nos locais de 1 a 5, respectivamente.

As 988 árvores cubadas foram reunidas em uma única planilha eletrônica. Para diferenciar os locais de coleta, cinco variáveis *dummy* foram criadas, de forma que quando o local era igual a 1, a variável *dummy* 1 assumia valor 1, para os demais locais, *dummy* 1 assumia valor 0. A mesma regra vale para as outras variáveis *dummy*. O vetor utilizado foi definido como  $D = [D1 \ D2 \ D3 \ D4 \ D5]$ , sendo  $D_i=1$  e  $D_i'=0$ ,  $i=1, 2, 3, 4, 5$  e  $i <> i'$ .

Os *inputs* da rede foram gerados ao multiplicar *dap* e *ht* pelas variáveis *dummy*. Desta forma, foram obtidas dez variáveis de entrada, das quais cinco referentes ao *dap* e cinco referentes a *ht*.

Embora houvesse espécies diferentes, não foi necessário atribuir variável para a mesma, pois o local já indicava indiretamente a espécie plantada. Caso esta diferenciação não ocorresse, uma variável categórica deveria ser acrescentada.

Depois de estruturado o banco de dados, foram aplicados três pré-processamentos diferentes (Tabela 1), gerando três novos arquivos de dados. O pré-processamento 1 apenas normalizou os valores entre 0 e 1. Já o 2 e 3, além de normalizar, resultam em uma equalização dos dados entre 0 e 0,8, e entre 0,1 e 0,8, respectivamente. A normalização evita a polarização da rede e a equalização dos dados, a saturação das funções de ativação.

Tabela 1 – Análise descritiva dos dados de cada local

Local	Estatística	<i>dap</i>	<i>alt</i>	<i>volume</i>
SP	Média	9,7155	9,0860	0,0380
	Desvio-padrão	2,5106	2,0043	0,0223
	Mínimo	3,0000	3,4000	0,0015
	Máximo	15,6000	13,4000	0,1094
BA	Média	18,0521	24,0611	0,3413
	Desvio-padrão	6,4886	5,2191	0,2495
	Mínimo	7,0000	11,5000	0,0231
	Máximo	31,0000	35,0000	1,0093
MG1	Média	24,6673	27,2127	0,5664
	Desvio-padrão	5,5074	6,2084	0,3098
	Mínimo	16,5000	16,7000	0,1472
	Máximo	34,4000	36,3000	1,1093
MG2	Média	18,1527	27,3071	0,4126
	Desvio-padrão	6,2939	6,1370	0,3719
	Mínimo	8,6500	15,7800	0,0476
	Máximo	37,7500	47,6100	2,3597
MT	Média	9,8667	8,0633	0,0344
	Desvio-padrão	2,8660	1,9049	0,0195
	Mínimo	5,7000	5,1000	0,0108
	Máximo	15,9000	12,2000	0,0766

Tabela 2 – Pré-processamentos aplicados aos dados

Pré-Processamento	Fórmula
1	valor transformado = $1 + \left( \frac{-x + \text{Mínimo}}{\text{Máximo}} \right)$
2	valor transformado = $0,8 + \left( \frac{-x + \text{Mínimo}}{\text{Máximo}} \right)$
3	valor transformado = $0,1 + \left( \frac{0,8 * (x - \text{Mínimo})}{(\text{Máximo} - \text{Mínimo})} \right)$

Após a eliminação dos *outliers*, os dados foram separados de forma aleatória em dois conjuntos. O de treinamento, composto por 60% dos dados, e o de validação, composto pelos dados complementares (40%).

Foram construídas redes com três arquiteturas diferentes (Tabela 3, Figura 1), todas usando a função de ativação logística. Os pesos iniciais foram gerados de forma aleatória entre os valores -0,5 a 0,5. As camadas foram interligadas por sinapses completas, ou seja, cada neurônio da camada *i* foi ligado a todos os neurônios da camada seguinte (*i+1*), sendo as sinapses orientadas somente para frente (redes *feedforward*).

O algoritmo de aprendizado utilizado foi o da retropropagação do erro (*backpropagation*) e os parâmetros de treinamento foram taxa de aprendizagem de 0,2 e termo *momentum* de 0,9. O *software* utilizado para o treinamento foi o JavaSNNS (ZELL *et al.*, 1998) (disponível em: <<http://www-ra.informatik.uni-tuebingen.de/SNNS/>>). O treinamento foi encerrado quando o gráfico de resíduos tendia à estabilização.

As redes foram avaliadas quanto à eficiência, por meio do teste *t*, comparando valores estimados e valores observados para o conjunto de validação. As redes que resultaram em diferenças não-significativas, em relação aos valores observados, foram analisadas quanto à distribuição dos resíduos. Os dados processados pela rede foram submetidos a pós-processamento, composto por uma regra lógica para zerar todos os valores negativos. Foram também submetidos à operação inversa do pré-processamento, visando restaurar a escala original dos dados.

Tabela 3 – Redes construídas com diferentes arquiteturas e função de ativação logística

Arquitetura	1	2	3
Função de Ativação	Logística	Logística	Logística
Camada oculta 1	5	10	10
Camada oculta 2	5	-	8

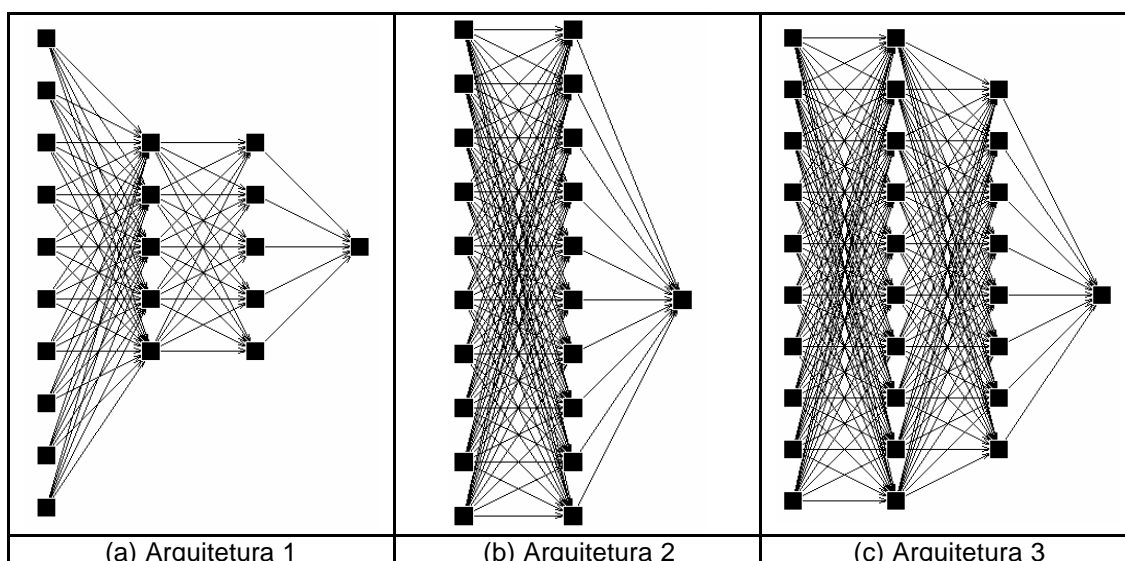


Figura 1 – Arquiteturas das redes utilizadas.

### 3. Resultados e Discussão

Os bancos de dados formados pelos diferentes pré-processamentos foram usados para treinar e validar a rede. Como são três arquiteturas diferentes e três formas de pré-processamento, foram obtidas nove redes ajustadas (Tabela 4).

Tabela 4 – Identificação das nove redes obtidas após os treinamentos

Pré-Processamento/Arquitetura	1	2	3
1	40	44	48
2	41	45	49
3	42	46	50



Após o treinamento, os dados de validação foram então processados e comparados aos valores observados pelo teste t. Para o nível de significância de 5%, as redes 41, 42, 45, 46, 48, 49 e 50 apresentaram valores estimados estatisticamente iguais aos observados. Estas redes então foram avaliadas quanto à distribuição dos resíduos (Figura 2).

Por meio dos gráficos de resíduos pode-se notar que, embora significativamente iguais pelo teste t, algumas redes apresentaram tendências indesejáveis. As redes 42 e 49 apresentaram tendência de superestimar valores baixos de volume (árvores de pequeno porte).

As redes 41 e 45 resultaram, para árvores de pequeno porte, em “bifurcação” dos dados: ou ocorre a superestimação dos valores, ou a subestimação de outros, causando no gráfico um vazio na região por onde passa a reta de 45°. Quanto mais sobrepostos a esta reta, mais eficiente é a rede em estimar os valores de volume.

Os gráficos de resíduos das redes 46, 48 e 50 apresentaram comportamento adequado, sem tendenciosidade na estimação.

Após as análises gráficas, as redes que se apresentaram adequadas ao uso foram 46, 48 e 50 (Figuras 2 e 3). As redes 46 e 50 foram treinadas com dados do pré-processamento 3. Por sua vez, a rede 48 utilizou dados pré-processados pelo método 1 para o treinamento.

Analisando a arquitetura, as redes 48 e 50 foram redes de duas camadas intermediárias, com dez e oito neurônios, respectivamente. Já a rede 46 possuía uma única camada intermediária de dez neurônios.

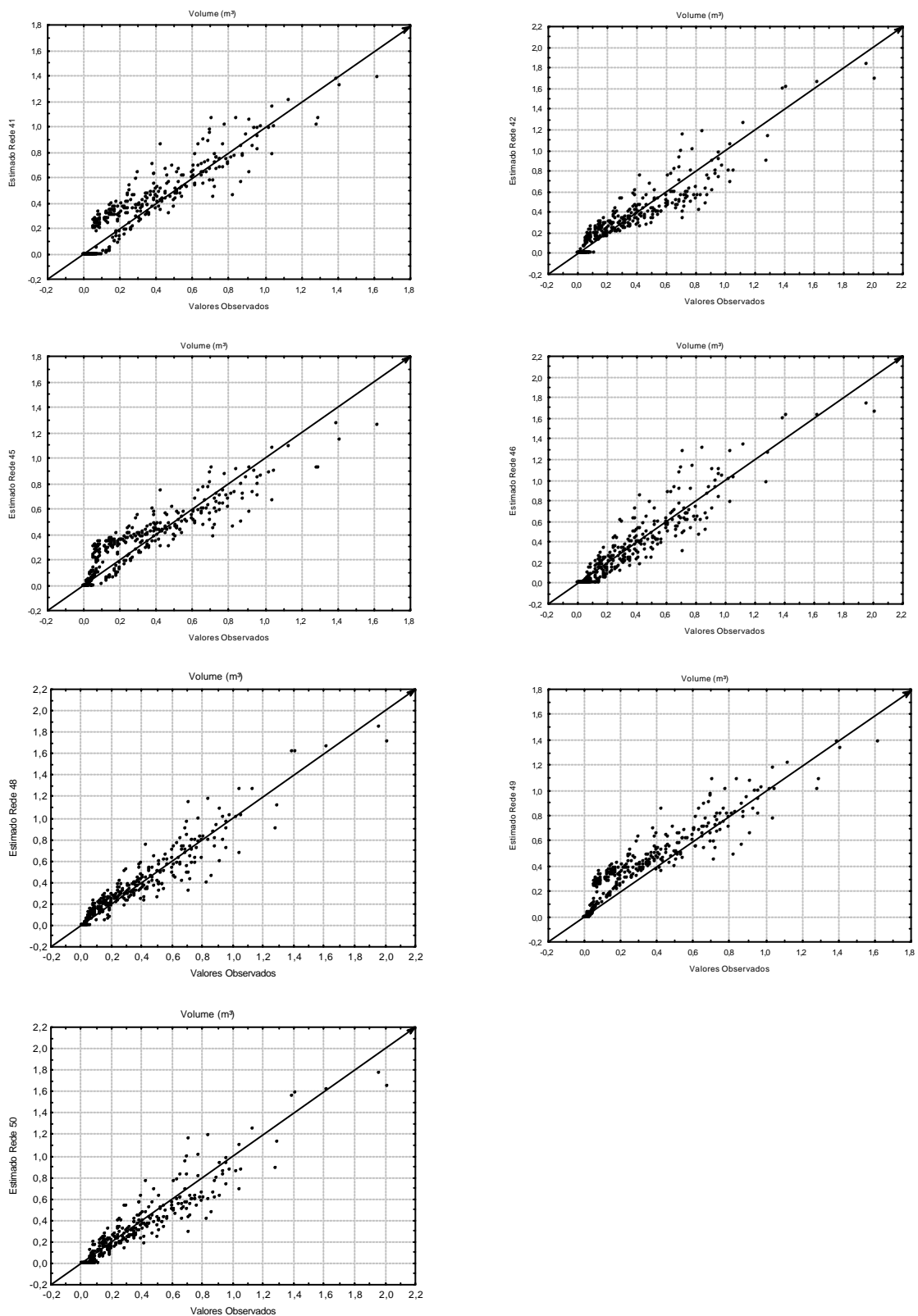
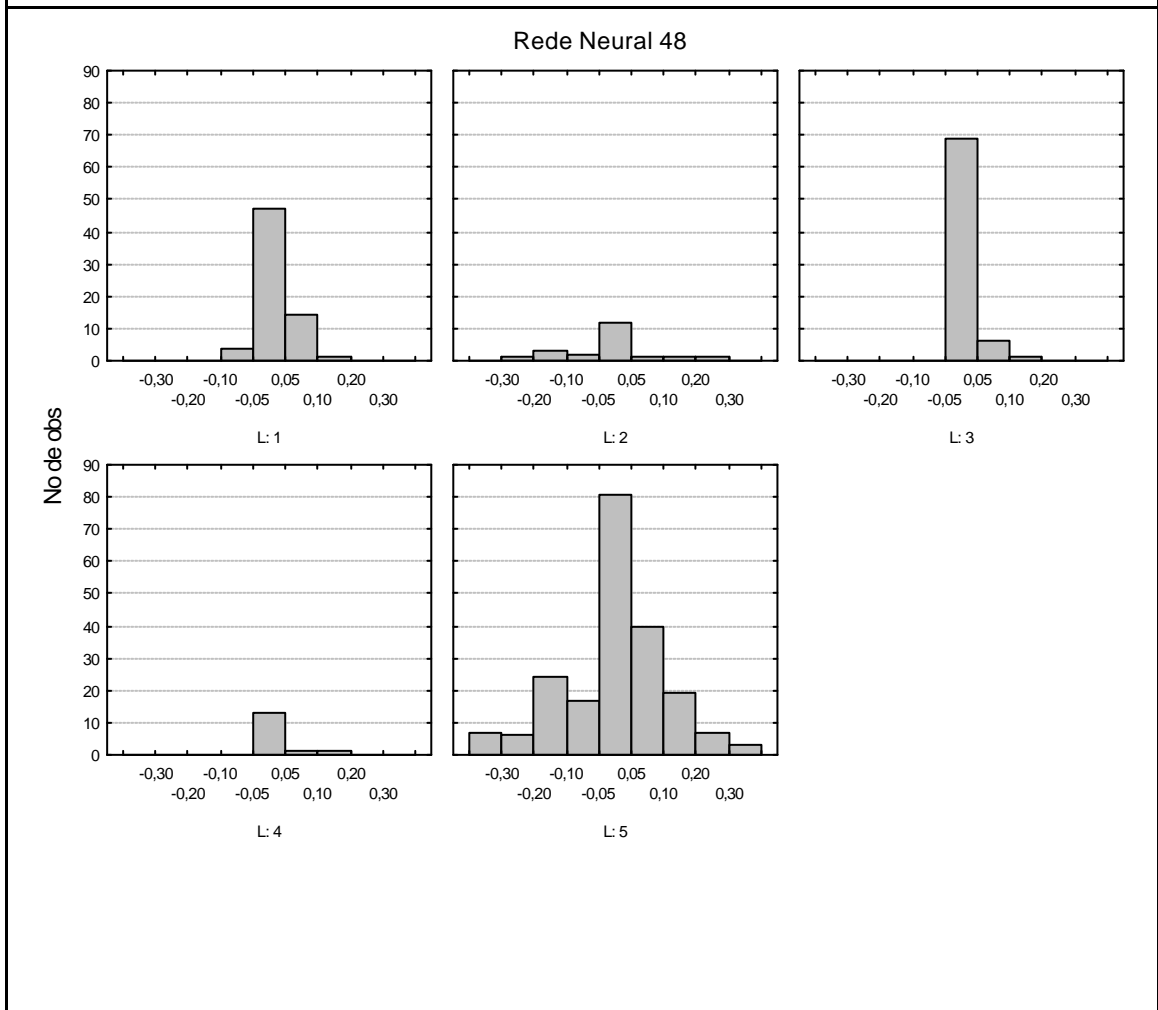
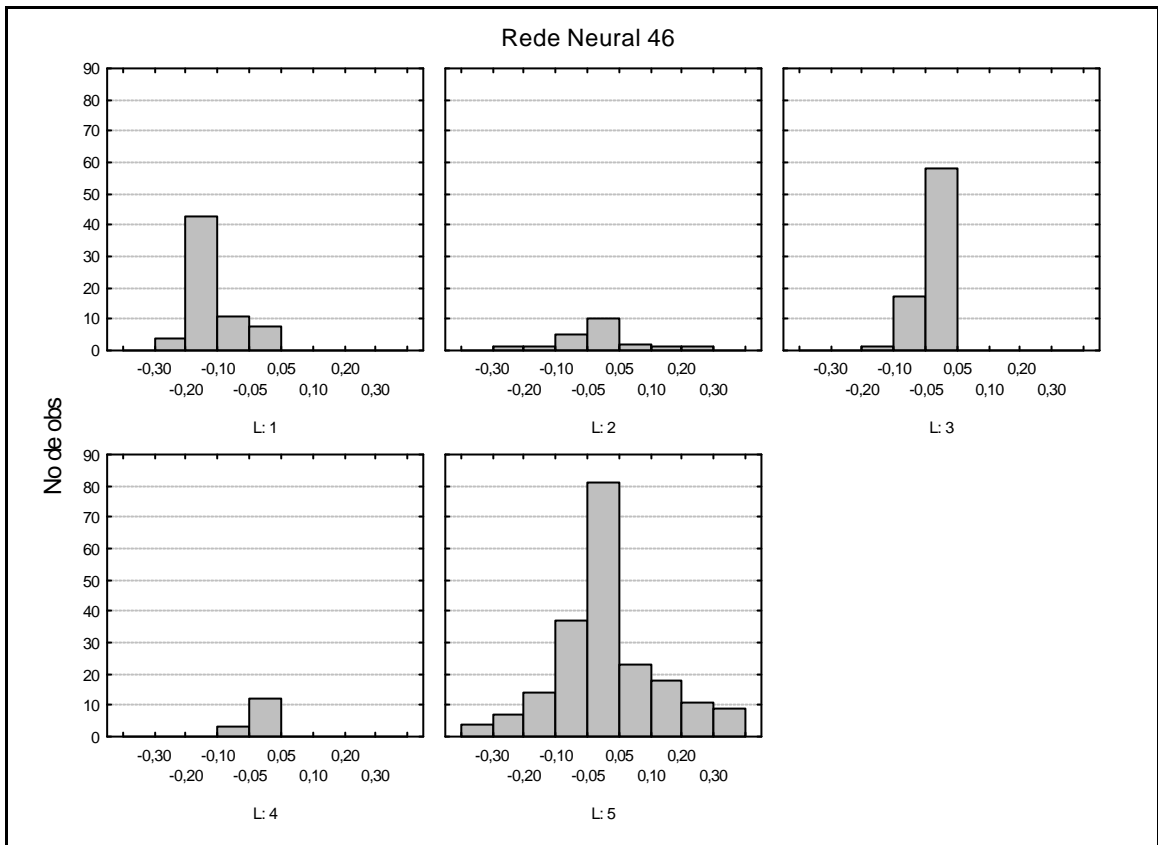


Figura 2 – Gráficos de resíduos para as redes que apresentaram valores estimados estatisticamente iguais aos valores observados.



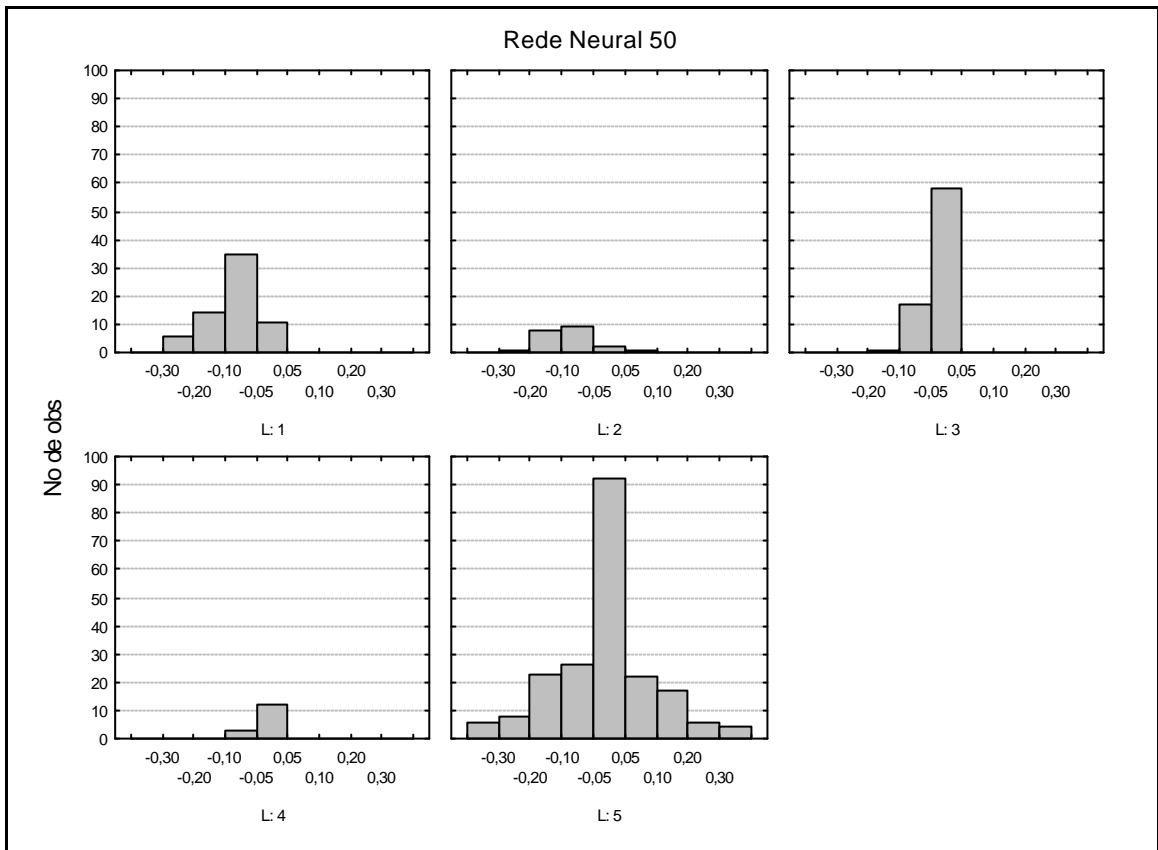


Figura 3 – Distribuição dos resíduos das estimativas de volume geradas pelas redes 46, 48 e 50, para os locais 1, 2, 3, 4 e 5.

#### 4. Conclusões

Com base nos resultados, verifica-se que a arquitetura 1 não apresentou nenhuma rede adequada. O mesmo ocorreu para o pré-processamento 2, já que nenhuma rede treinada com os dados desse gerou resultados eficientes e livres de bias. Tanto o pré-processamento 2 quanto o 3, além da normalização, também realizam a equalização dos dados. O pré-processamento mais eficiente foi o de número 3, seguido pelo pré-processamento 1. Analisando as três diferentes arquiteturas, a que apresentou melhor desempenho foi a arquitetura 3, seguida pela arquitetura 2.

Assim, foi possível concluir que:

- Dados de árvores cubadas devem ser pré-processados tanto para normalização quanto para equalização.

- Para problemas de aproximação de funções de estimação de volume de árvores devem ser utilizadas redes com duas camadas intermediárias, com dez e oito neurônios, respectivamente.

- Redes neurais artificiais podem ser utilizadas para estimar o volume de árvores.

## 5. Referências Bibliográficas

BISHOP, C. M. **Neural networks for pattern recognition**. Oxford: Oxford University Press, 1995. 482 p.

BRAGA, A. de P.; CARVALHO, A. P. de L. F. de; LUDEMIR, T. B. **Fundamentos de redes neurais artificiais**. Rio de Janeiro: DCC/I, COPPE/Sistemas, NCE/UFRJ, 1998. 246 p.

GOLDSCHIMDT, R.; PASSOS, E. **Data mining: um guia prático**. Rio de Janeiro: Elsevier, 2005. 261 p.

HAYKIN, S. **Redes neurais: princípios e prática**. 2. ed. Porto Alegre: Bookman, 2001. 900 p.

KAVZOGLU, T.; MATHER, P. M. The use of backpropagating artificial neural networks in land cover classification. **Int. J. Remote Sensing**, v. 24, n. 23, p. 4907-4938, 2003.

KOVÁCS, Z. L. **Redes neurais artificiais: fundamentos e aplicações**. 2. ed. Ver. e amp. São Paulo: Colledium cognitio, 1996. 174 p.

ZELL, A.; MAMIER, G.; MACHE, M. V. N.; HÜBNER, R.; DÖRIN, S.; HERMANN, K-U; SOYEZ, T.; SCHMALZL, M.; SOMMER, T.; HATZIGEORGIOU, A.; POSSELT, D.; SCHREINER, T.; KETT, B.; CLEMENTE, G.; WIELAND, J.; GATTER, J. **SNNS – Stuttgart Neural Network Simulator v.4.2**, User Manual, University of Stuttgart/University of Tübingen, 1998. 338 p.

## 4. CONCLUSÃO GERAL

A introdução apresenta a técnica de RNAs de forma simples e objetiva, além de situar o leitor junto ao desafio proposto para este trabalho.

O material apresentado nos capítulos iniciais deste texto permitiu a imersão na teoria de redes neurais artificiais de forma sintetizada e ordenada.

Por meio dos métodos de manipulação de banco de dados, pode-se melhorar de maneira significativa o treinamento de RNA.

O pré-processamento correto dos dados foi imprescindível para obter resultados de qualidade.

Diferentes arquiteturas podem ser utilizadas com sucesso, permitindo ao usuário bastante flexibilidade.

As redes neurais artificiais se mostraram eficientes na estimação de volume de árvores.

O *software* utilizado atendeu às necessidades deste trabalho. Alguns bugs precisam ser contornados pelo uso de diferentes versões, ao longo de um projeto de RNAs. A existência de grande quantidade de funções de ativação e de algoritmos de treinamentos torna o *software* bastante flexível.

## REFERÊNCIAS BIBLIOGRÁFICAS

BISHOP, C. M. **Neural networks for pattern recognition**. Oxford: Oxford University Press, 1995. 482 p.

BRAGA, A. de P.; CARVALHO, A. P. de L. F. de; LUDEMIR, T. B. **Fundamentos de redes neurais artificiais**. Rio de Janeiro: DCC/I,, COPPE/Sistemas, NCE/UFRJ, 1998. 246 p.

BULLINARIA, J. A. Apresenta informações sobre Introduction to neural network. Disponível em <<http://www.cs.bham.ac.uk/~jxb/inn.html>>. Acesso em: [ca. 2000].

CAMPOS, J. C. C.; LEITE, H. G. **Mensuração floretal**: perguntas e respostas. 2. ed. rev. e ampl. Viçosa: UFV, 2006. 470 p.

GOLDSCHMIDT, R.; PASSOS, E. **Data mining**: um guia prático. Rio de Janeiro: Elsevier, 2005. 261 p.

HAIR JR., J. F.; ANDERSON, R. E.; TATHAM, R. L.; BLACK, W. C. **Análise multivariada de dados**. Porto Alegre: Bookman, 2005.

HAYKIN, S. **Redes neurais**: princípios e prática. 2. ed. Porto Alegre: Bookman, 2001. 900 p.

KOVÁCS, Z. L. **Redes neurais artificiais**: fundamentos e aplicações. 2. ed. Ver. e amp. São Paulo: Colledium cognitio, 1996. 174 p.

LEITE, H. G.; ANDRADE, V. C. L. de. Um método para condução de inventários florestais sem o uso de equações volumétricas. **Rev. Árvore**, v. 26, n. 3, p. 321-328, 2002.

LEITE, H. G.; GUIMARÃES, D. P.; CAMPOS, J. C. C. Descrição e emprego de um modelo para estimar múltiplos volumes de árvores. **Rev. Árvore**, v. 19, n. 1, p. 1-21, 1995.

MATEUS, G. R.; LUNA, H. P. L. **Programação Não Linear**. V Escola de Computação. Belo Horizonte: UFMG, 1986. 289 p.

NEURAL NETWORK FAQ: Introduction. Desenvolvido por Sarle, W. S., 1997-. Apresenta informações sobre redes neurais. Disponível em <tp://ftp.sas.com/pub/neural/FAQ.html> Acesso em: 15 maio 2005.

SCHUMACHER, F.X.; HALL, F.S. Logarithmic expression of timber-tree volume. **Journal of Agricultural Research**, v. 47, n. 9, p. 719-734, 1933.

VON ZUBEN, F.J. Uma caricatura funcional de redes neurais artificiais. **Learning and Nonlinear Models**, v. 1, n. 2, p. 77-87, 2003.



## APÊNDICES

## APÊNDICE A

SNNS é um *software* de simulação de Redes Neurais desenvolvido pelo *Institute for Parallel and Distributed High Performance Systems* na Universidade de Stuttgart. O objetivo do SNNS é criar um ambiente de simulação eficiente e flexível para pesquisa na área de redes neurais.

Atualmente, as arquiteturas e os procedimentos de aprendizado presentes no *software* são:

- *Backpropagation (BP) for feedforward networks*
  - *vanilla (online) BP*
  - *BP with momentum term and flat spot elimination*
  - *batch BP*
- *Counterpropagation*
- *Quickprop*
- *Backpercolation 1*
- *RProp*
- *Generalized radial basis functions (RBF)*
- *ART1*
- *ART2*
- *ARTMAP*
- *Cascade Correlation*
- *Recurrent Cascade Correlation*
- *Dynamic LVQ*
- *Backpropagation through time (for recurrent networks)*
- *Quickprop through time (for recurrent networks)*
- *Self-organizing maps (Kohonen maps)*
- *TDNN (time-delay networks) with Backpropagation*
- *Jordan networks*
- *Elman networks and extended hierarchical Elman networks*
- *Associative Memory*

Existem três versões do programa disponível para *download*. As duas primeiras versões, SNNS 4.1 e 4.2, são desenvolvidas para plataforma UNIX. Já a versão JavaNNS funciona na máquina JAVA.

Na página do *software* (disponível em: <<http://www-ra.informatik.uni-tuebingen.de/SNNS/>>), o usuário encontra um fórum para esclarecimento de dúvidas, além dos manuais em formato digital para consultas *on-line*.

Embora desenvolvido por uma universidade alemã, o *software* possui os comandos e as janelas em inglês, o que o coloca entre os mais utilizados mundialmente.

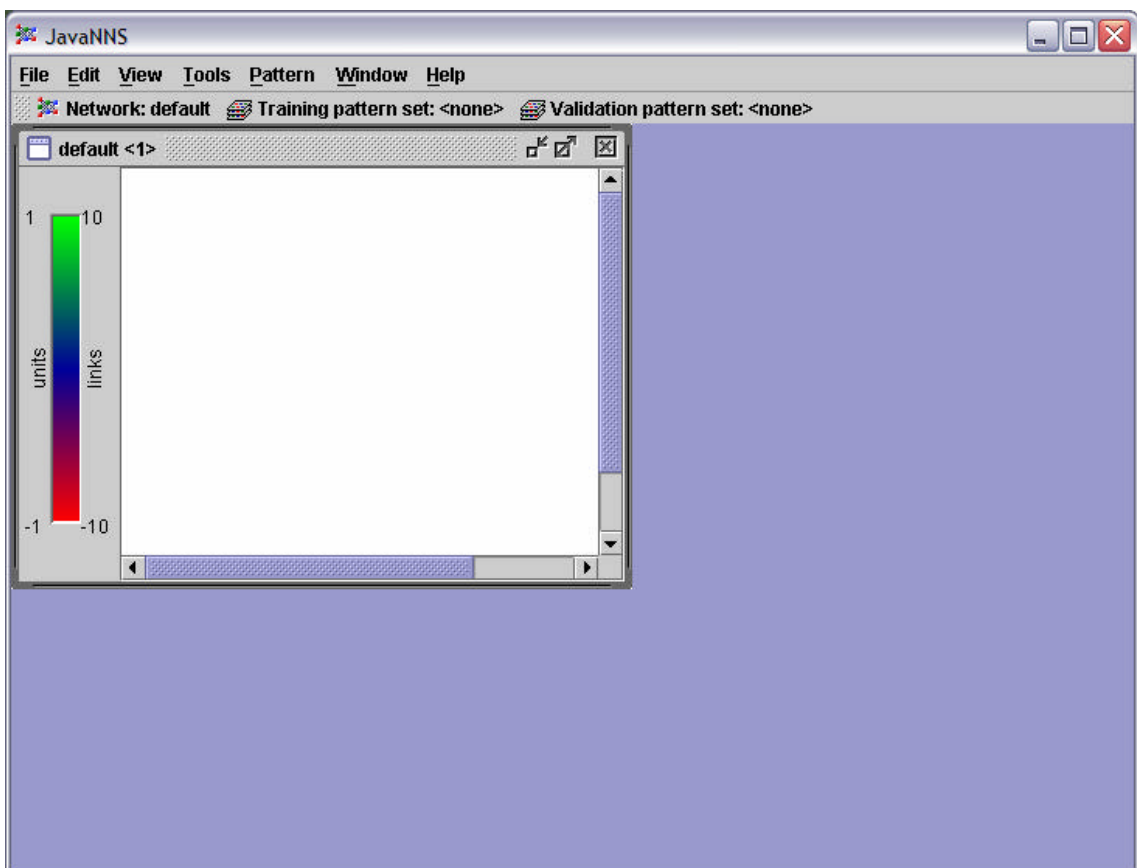


Figura 1 – Ambiente de trabalho inicial do *software* JavaNNS.

**Apresentação dos dados.** Os dados podem ser apresentados ao *software* padrão por padrão. Primeiramente os *inputs*, e na seqüência os *outputs*. Quando se trabalha com grande volume de dados, deve-se preferir a apresentação matricial dos dados. Um exemplo do arquivo em formato matricial pode ser visto no Apêndice.

Vale ressaltar que o cabeçalho é padrão, devendo o usuário modificar apenas as dimensões da matriz de entrada e saída dos dados. Ao carregar o arquivo de dados no programa, a existência de qualquer erro no arquivo causará recusa imediata.

**Construindo uma rede.** O software SNNS possui uma janela onde a rede será apresentada visualmente, através de quadrados e linhas. Primeiramente constroem-se as camadas e em seguida ligam-se os neurônios.

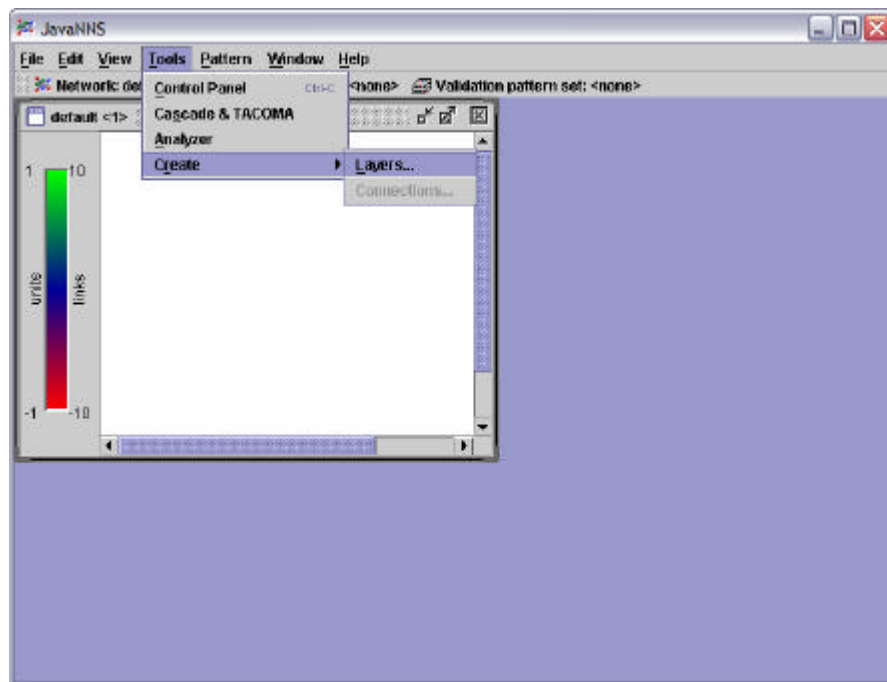


Figura 2 – Comando para a criação de camadas de uma rede neural.

A inclusão de neurônios é feita camada a camada. Na primeira parte da janela de criação, o usuário determina a altura e a largura da camada. A altura é a quantidade de neurônios desta camada. Por exemplo, se o usuário definir a altura (*Height*) como sendo 10, a camada em questão terá dez neurônios.

*Top left position* consiste na posição do espaço (janela) em que o primeiro neurônio da camada em questão será posicionado.

Na seqüência, define-se o tipo de nodo que se está criando, por exemplo, neurônios de entrada. Também, determina a função de ativação dos neurônios e, quando houver necessidade, a função de saída. Automaticamente o programa informa qual a camada que está sendo construída no momento (*Layer number*).

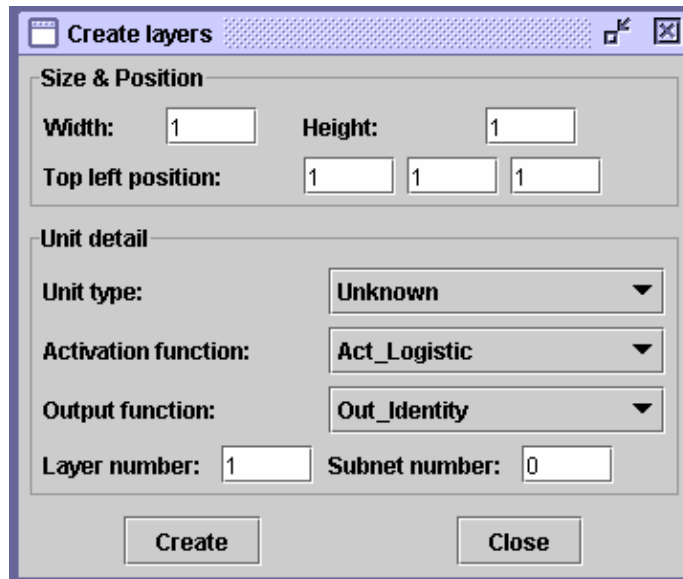


Figura 3 – Janela de criação de camadas.

Depois de criadas as camadas com seus respectivos neurônios, são feitas as conexões entre eles.

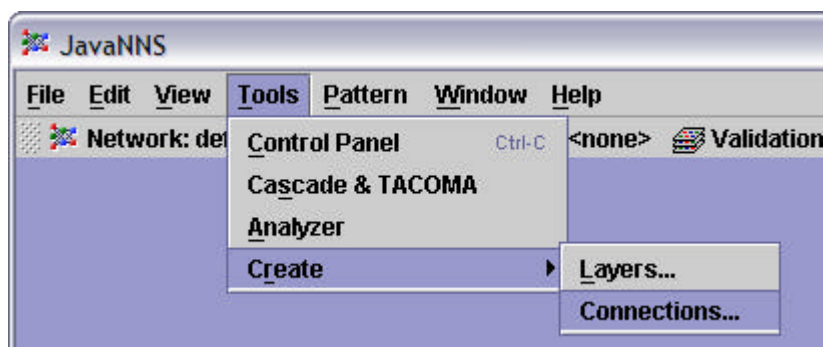


Figura 4 – Comando para a criação de conexões entre neurônios.

**Treinamento.** Esta é sem dúvida a etapa mais esperada por todos os usuários da técnica de redes neurais, como também a etapa mais subjetiva de todo o processo. O grande número de parâmetros a serem definidos faz com que existam diversas combinações.

No *software*, todo o treinamento é tratado na janela denominada Painel de Controle (*Control Painel*). Na aba *Initializing* defini-se a forma de determinação dos pesos iniciais. O *default* é a determinação de forma aleatória dentro de um intervalo definido pelo usuário.

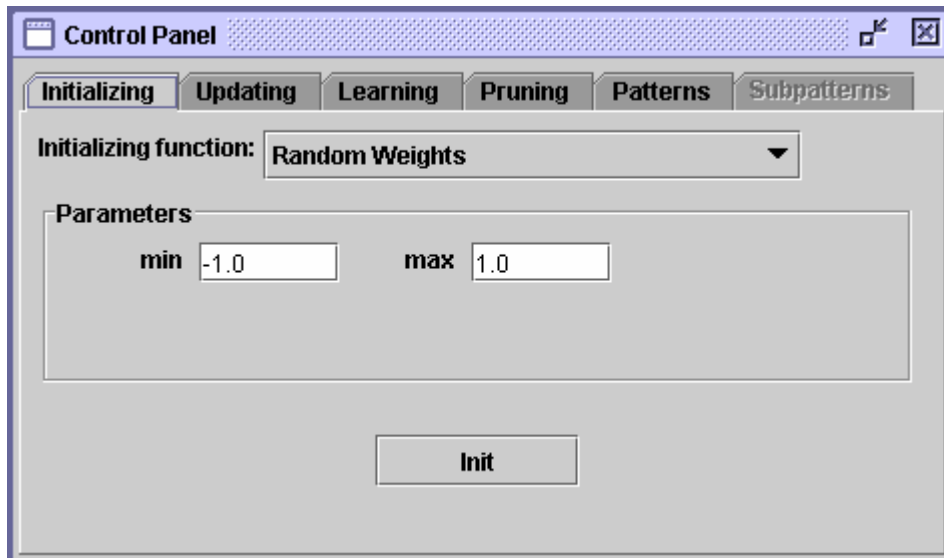


Figura 5 – Forma com que os pesos serão inicializados.

Na aba *Updating*, o usuário define a ordem com que os pesos serão ajustados. Recomenda-se deixar a escolha *default* do programa.

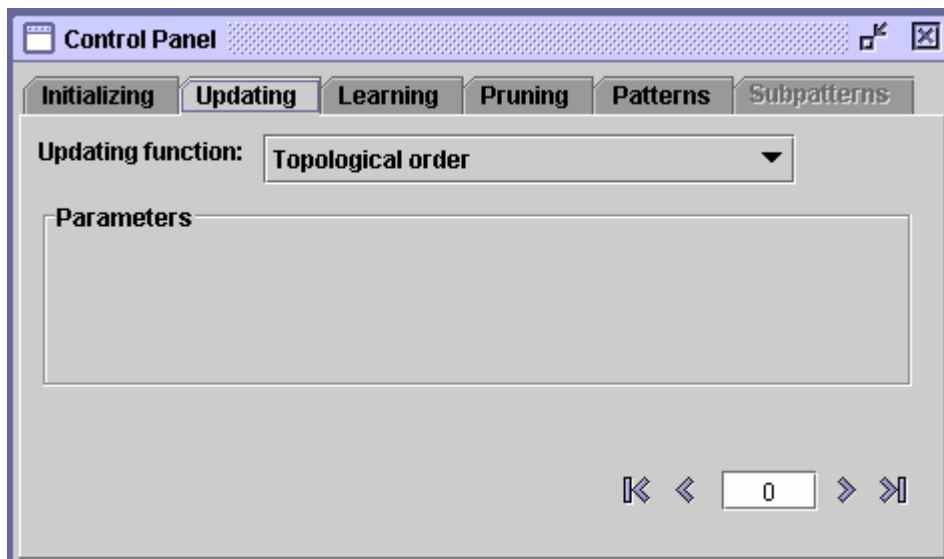


Figura 6 – Janela para determinar o esquema de atualização dos pesos da rede.

Em *Learning*, o algoritmo de treinamento da rede pode ser escolhido, assim como seus respectivos parâmetros. No exemplo, vemos que para o algoritmo de aprendizagem *backpropagation* temos dois parâmetros que

podem ser alterados pelo usuário. Também nesta janela, define-se o número de ciclos e os passos de atualização dos pesos. Para maiores informações sobre os parâmetros, o manual do *software* deve ser consultado.

Se for marcada a opção *Shuffle*, os dados serão apresentados à rede de forma aleatória.

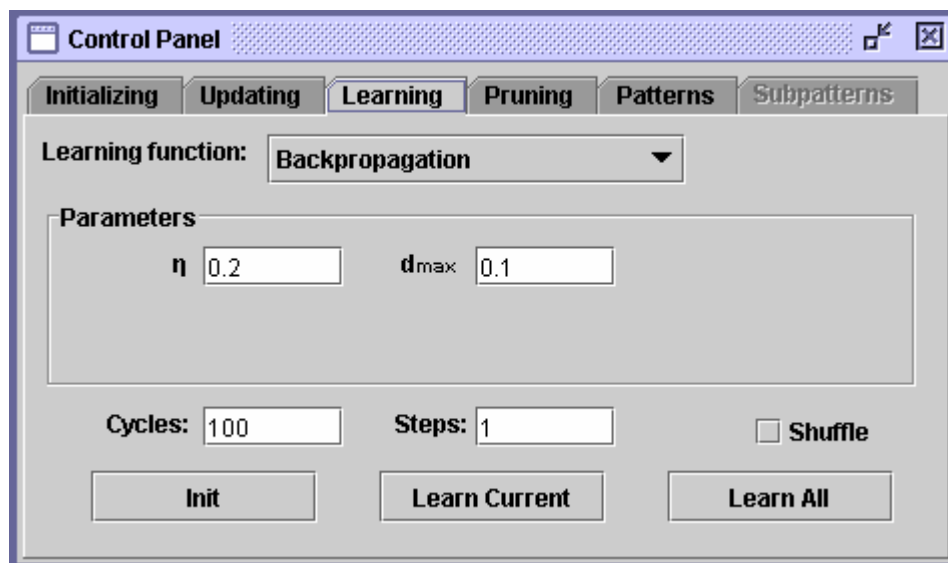


Figura 7 – Algoritmo de treinamento e seus respectivos parâmetros. Definição do número de ciclos e do número de passos para atualização dos pesos.

A aba *Patterns* é utilizada para definir os arquivos que serão utilizados para treinamento e para validação. Ressalta-se que estes devem ser abertos, primeiramente, via menu principal, e depois de abertos, devem ser definidos como *training set* ou *validation set*.

Quando for o caso, é nessa janela que se determina a função de remapeamento. Por não ser usada neste trabalho, para maiores detalhes recomenda-se consultar o manual.

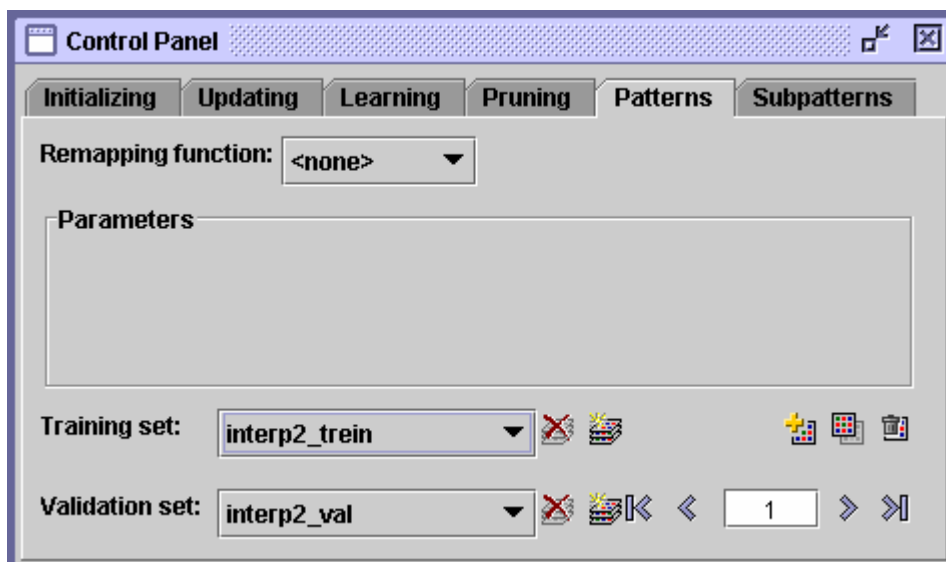


Figura 8 – Estabelece os arquivos de treinamento e validação.

Quando os dados forem apresentados de forma matricial, deve-se definir o modo com que o programa irá fazer a leitura da matriz de informações. É na aba *subpatterns* que se faz esta definição. Para apresentação aleatória dos dados, marque a opção *Shuffle*. No exemplo em questão, os dados de entrada na dimensão 2 devem ser lidos de nove em nove. Logo, o tamanho (*size*) e o passo (*step*) devem ser 9.

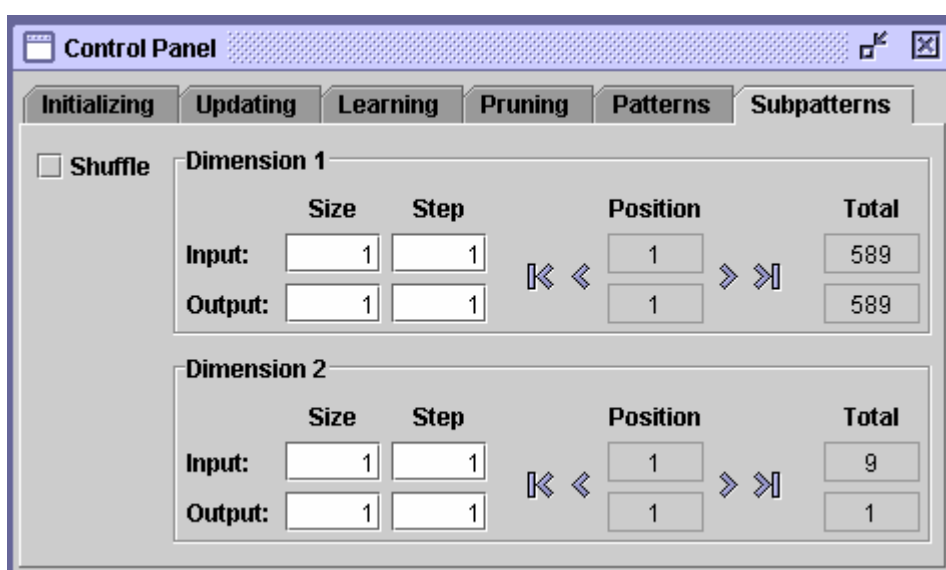


Figura 9 – Janela para orientação da leitura de arquivos sob a forma matricial.



**Gráfico de erro.** O treinamento e a validação podem ser acompanhados pelo gráfico de erro. Para acessá-lo basta seguir os passos abaixo.

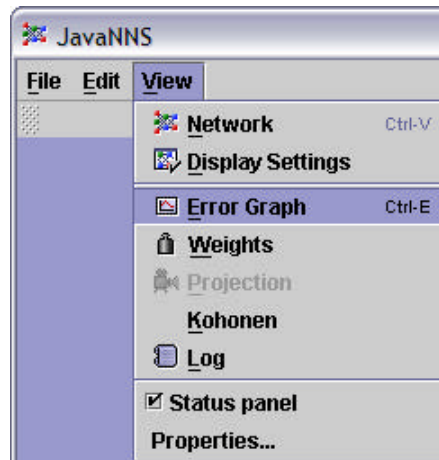


Figura 10 – Comando para ativar a visualização do gráfico de erros.

Como não há um critério muito claro para encerrar o treinamento de uma rede, este gráfico possui uma grande importância. Quando a curva do erro torna-se constante, recomenda-se encerrar o treinamento. No exemplo a seguir, o treinamento deveria ter sido encerrado por volta do ciclo 100.

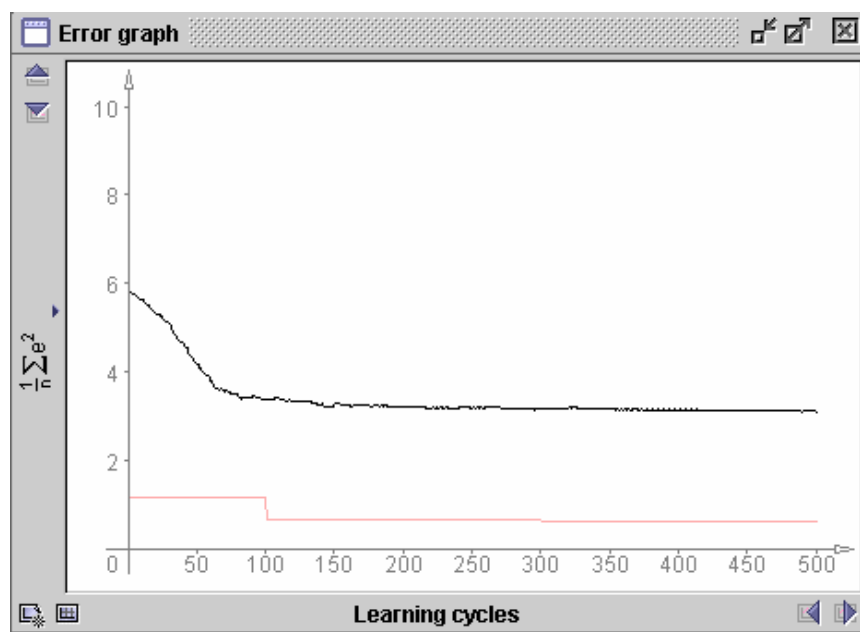


Figura 11 – Gráfico de erro médio por padrão apresentado. Curva do erro para treinamento e para validação.

Após o treinamento, a rede terá os seus pesos treinados e estará pronta para ser usada. Cada conexão entre neurônio terá um peso, e este peso será o mesmo até que a rede seja novamente treinada.

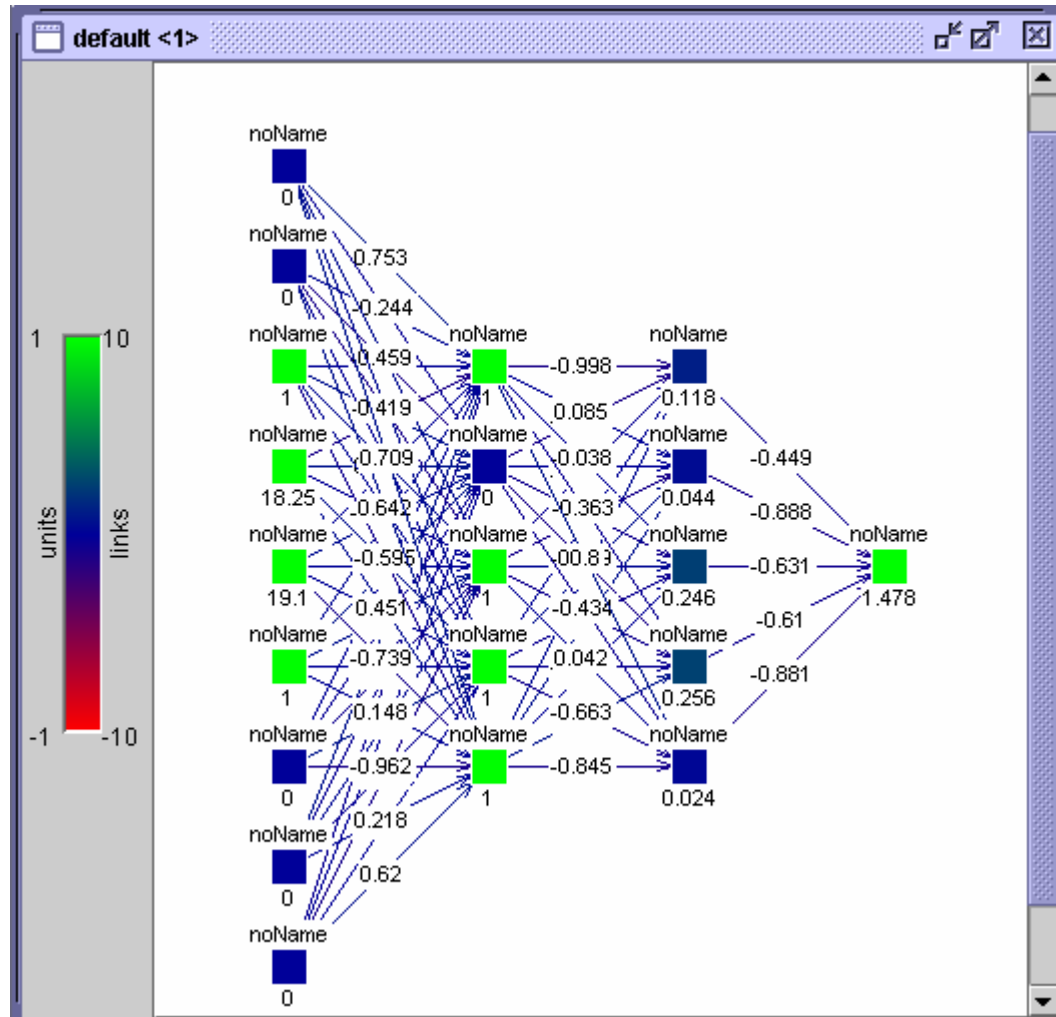


Figura 12 – Visualização da rede construída com os pesos ajustados.

Para rodar um conjunto de dados na rede treinada, o usuário deve utilizar a versão SNNS 4.2. Nesta versão, em UNIX, o usuário irá abrir o arquivo da rede (.net) e, em seguida, o arquivo que contém o conjunto dos dados a serem apresentados (.pat). Determina-se o padrão de leitura do arquivo. Para ter as saídas, basta salvar o arquivo resultado (.res).

## APÊNDICE B

### ARQUIVO EXEMPLO PARA ENTRADA DE DADOS NA FORMA MATRICIAL

SNNS pattern definition file v3.2 generated at 6/3/2006

No. of patterns : 1  
No. of input units : 1  
No. of output units : 1  
No. of variable input dimensions : 2  
Maximum input dimensions : [ 381 10 ]  
No. of variable output dimensions : 2  
Maximum output dimensions: [ 381 1 ]

#Pattern 1 (Input)

[ 381 10 ]

0.50500	0.00000	0.00000	0.00000	0.00000
0.37800	0.00000	0.00000	0.00000	0.00000
0.25750	0.00000	0.00000	0.00000	0.00000
0.27000	0.00000	0.00000	0.00000	0.00000
0.71250	0.00000	0.00000	0.00000	0.00000
0.58600	0.00000	0.00000	0.00000	0.00000
0.57750	0.00000	0.00000	0.00000	0.00000
0.47800	0.00000	0.00000	0.00000	0.00000
0.43250	0.00000	0.00000	0.00000	0.00000
0.39200	0.00000	0.00000	0.00000	0.00000
0.22250	0.00000	0.00000	0.00000	0.00000
0.33800	0.00000	0.00000	0.00000	0.00000
0.68500	0.00000	0.00000	0.00000	0.00000
0.62600	0.00000	0.00000	0.00000	0.00000
0.71500	0.00000	0.00000	0.00000	0.00000
0.57400	0.00000	0.00000	0.00000	0.00000
0.30250	0.00000	0.00000	0.00000	0.00000
0.27200	0.00000	0.00000	0.00000	0.00000
0.63500	0.00000	0.00000	0.00000	0.00000
0.44800	0.00000	0.00000	0.00000	0.00000
0.40250	0.00000	0.00000	0.00000	0.00000
0.30000	0.00000	0.00000	0.00000	0.00000
0.35000	0.00000	0.00000	0.00000	0.00000
0.51700	0.00000	0.00000	0.00000	0.00000
0.56750	0.00000	0.00000	0.00000	0.00000
0.51100	0.00000	0.00000	0.00000	0.00000
0.31250	0.00000	0.00000	0.00000	0.00000
0.41100	0.00000	0.00000	0.00000	0.00000
0.44750	0.00000	0.00000	0.00000	0.00000
0.44600	0.00000	0.00000	0.00000	0.00000
0.39250	0.00000	0.00000	0.00000	0.00000
0.44000	0.00000	0.00000	0.00000	0.00000
0.48500	0.00000	0.00000	0.00000	0.00000
0.43800	0.00000	0.00000	0.00000	0.00000
0.57000	0.00000	0.00000	0.00000	0.00000
0.44600	0.00000	0.00000	0.00000	0.00000
0.39500	0.00000	0.00000	0.00000	0.00000

0.42000	0.00000	0.00000	0.00000	0.00000
0.29500	0.00000	0.00000	0.00000	0.00000
0.37800	0.00000	0.00000	0.00000	0.00000
0.60750	0.00000	0.00000	0.00000	0.00000
0.44000	0.00000	0.00000	0.00000	0.00000
0.56750	0.00000	0.00000	0.00000	0.00000
0.41200	0.00000	0.00000	0.00000	0.00000
0.47500	0.00000	0.00000	0.00000	0.00000
0.36400	0.00000	0.00000	0.00000	0.00000
0.39000	0.00000	0.00000	0.00000	0.00000
0.33600	0.00000	0.00000	0.00000	0.00000
0.50750	0.00000	0.00000	0.00000	0.00000
0.39200	0.00000	0.00000	0.00000	0.00000
0.63000	0.00000	0.00000	0.00000	0.00000
0.54000	0.00000	0.00000	0.00000	0.00000
0.52750	0.00000	0.00000	0.00000	0.00000
0.37600	0.00000	0.00000	0.00000	0.00000
0.20500	0.00000	0.00000	0.00000	0.00000
0.31800	0.00000	0.00000	0.00000	0.00000
0.24750	0.00000	0.00000	0.00000	0.00000
0.32600	0.00000	0.00000	0.00000	0.00000
0.21250	0.00000	0.00000	0.00000	0.00000
0.30600	0.00000	0.00000	0.00000	0.00000
0.54750	0.00000	0.00000	0.00000	0.00000
0.45800	0.00000	0.00000	0.00000	0.00000
0.33500	0.00000	0.00000	0.00000	0.00000
0.32000	0.00000	0.00000	0.00000	0.00000
0.24250	0.00000	0.00000	0.00000	0.00000
0.26600	0.00000	0.00000	0.00000	0.00000
0.16750	0.00000	0.00000	0.00000	0.00000
0.31400	0.00000	0.00000	0.00000	0.00000
0.60750	0.00000	0.00000	0.00000	0.00000
0.47200	0.00000	0.00000	0.00000	0.00000
0.48000	0.00000	0.00000	0.00000	0.00000
0.35000	0.00000	0.00000	0.00000	0.00000
0.28500	0.00000	0.00000	0.00000	0.00000
0.32800	0.00000	0.00000	0.00000	0.00000
0.22250	0.00000	0.00000	0.00000	0.00000
0.30400	0.00000	0.00000	0.00000	0.00000
0.28500	0.00000	0.00000	0.00000	0.00000
0.30200	0.00000	0.00000	0.00000	0.00000
0.25250	0.00000	0.00000	0.00000	0.00000
0.25600	0.00000	0.00000	0.00000	0.00000
0.57250	0.00000	0.00000	0.00000	0.00000
0.48600	0.00000	0.00000	0.00000	0.00000
0.44750	0.00000	0.00000	0.00000	0.00000
0.35400	0.00000	0.00000	0.00000	0.00000
0.63250	0.00000	0.00000	0.00000	0.00000
0.46000	0.00000	0.00000	0.00000	0.00000
0.52000	0.00000	0.00000	0.00000	0.00000
0.38700	0.00000	0.00000	0.00000	0.00000
0.55000	0.00000	0.00000	0.00000	0.00000
0.35600	0.00000	0.00000	0.00000	0.00000
0.33500	0.00000	0.00000	0.00000	0.00000
0.32000	0.00000	0.00000	0.00000	0.00000
0.40250	0.00000	0.00000	0.00000	0.00000
0.37200	0.00000	0.00000	0.00000	0.00000
0.74750	0.00000	0.00000	0.00000	0.00000
0.67000	0.00000	0.00000	0.00000	0.00000
0.46000	0.00000	0.00000	0.00000	0.00000

0.35000	0.00000	0.00000	0.00000	0.00000
0.56250	0.00000	0.00000	0.00000	0.00000
0.38600	0.00000	0.00000	0.00000	0.00000
0.21500	0.00000	0.00000	0.00000	0.00000
0.31000	0.00000	0.00000	0.00000	0.00000
0.35500	0.00000	0.00000	0.00000	0.00000
0.40400	0.00000	0.00000	0.00000	0.00000
0.71750	0.00000	0.00000	0.00000	0.00000
0.60400	0.00000	0.00000	0.00000	0.00000
0.59750	0.00000	0.00000	0.00000	0.00000
0.48600	0.00000	0.00000	0.00000	0.00000
0.61000	0.00000	0.00000	0.00000	0.00000
0.46400	0.00000	0.00000	0.00000	0.00000
0.55250	0.00000	0.00000	0.00000	0.00000
0.44800	0.00000	0.00000	0.00000	0.00000
0.72250	0.00000	0.00000	0.00000	0.00000
0.54600	0.00000	0.00000	0.00000	0.00000
0.47250	0.00000	0.00000	0.00000	0.00000
0.46200	0.00000	0.00000	0.00000	0.00000
0.71750	0.00000	0.00000	0.00000	0.00000
0.57400	0.00000	0.00000	0.00000	0.00000
0.34250	0.00000	0.00000	0.00000	0.00000
0.34400	0.00000	0.00000	0.00000	0.00000
0.53000	0.00000	0.00000	0.00000	0.00000
0.41800	0.00000	0.00000	0.00000	0.00000
0.55000	0.00000	0.00000	0.00000	0.00000
0.39600	0.00000	0.00000	0.00000	0.00000
0.48000	0.00000	0.00000	0.00000	0.00000
0.36600	0.00000	0.00000	0.00000	0.00000
0.70750	0.00000	0.00000	0.00000	0.00000
0.54600	0.00000	0.00000	0.00000	0.00000
0.31000	0.00000	0.00000	0.00000	0.00000
0.43400	0.00000	0.00000	0.00000	0.00000
0.31250	0.00000	0.00000	0.00000	0.00000
0.27600	0.00000	0.00000	0.00000	0.00000
0.00000	0.47750	0.00000	0.00000	0.00000
0.00000	0.55600	0.00000	0.00000	0.00000
0.00000	0.44750	0.00000	0.00000	0.00000
0.00000	0.55600	0.00000	0.00000	0.00000
0.00000	0.46000	0.00000	0.00000	0.00000
0.00000	0.52400	0.00000	0.00000	0.00000
0.00000	0.47750	0.00000	0.00000	0.00000
0.00000	0.52600	0.00000	0.00000	0.00000
0.00000	0.46250	0.00000	0.00000	0.00000
0.00000	0.40200	0.00000	0.00000	0.00000
0.00000	0.36500	0.00000	0.00000	0.00000
0.00000	0.32200	0.00000	0.00000	0.00000
0.00000	0.31000	0.00000	0.00000	0.00000
0.00000	0.27000	0.00000	0.00000	0.00000
0.00000	0.24500	0.00000	0.00000	0.00000
0.00000	0.36200	0.00000	0.00000	0.00000
0.00000	0.31750	0.00000	0.00000	0.00000
0.00000	0.25600	0.00000	0.00000	0.00000
0.00000	0.30250	0.00000	0.00000	0.00000
0.00000	0.25000	0.00000	0.00000	0.00000
0.00000	0.30500	0.00000	0.00000	0.00000
0.00000	0.36000	0.00000	0.00000	0.00000
0.00000	0.25500	0.00000	0.00000	0.00000
0.00000	0.26000	0.00000	0.00000	0.00000
0.00000	0.28500	0.00000	0.00000	0.00000

0.00000	0.27800	0.00000	0.00000	0.00000
0.00000	0.27250	0.00000	0.00000	0.00000
0.00000	0.26200	0.00000	0.00000	0.00000
0.00000	0.19000	0.00000	0.00000	0.00000
0.00000	0.21400	0.00000	0.00000	0.00000
0.00000	0.07250	0.00000	0.00000	0.00000
0.00000	0.25000	0.00000	0.00000	0.00000
0.00000	0.18500	0.00000	0.00000	0.00000
0.00000	0.18400	0.00000	0.00000	0.00000
0.00000	0.07250	0.00000	0.00000	0.00000
0.00000	0.28400	0.00000	0.00000	0.00000
0.00000	0.14250	0.00000	0.00000	0.00000
0.00000	0.22400	0.00000	0.00000	0.00000
0.00000	0.06250	0.00000	0.00000	0.00000
0.00000	0.17400	0.00000	0.00000	0.00000
0.00000	0.15000	0.00000	0.00000	0.00000
0.00000	0.21000	0.00000	0.00000	0.00000
0.00000	0.00000	0.57750	0.00000	0.00000
0.00000	0.00000	0.67400	0.00000	0.00000
0.00000	0.00000	0.61500	0.00000	0.00000
0.00000	0.00000	0.70400	0.00000	0.00000
0.00000	0.00000	0.57750	0.00000	0.00000
0.00000	0.00000	0.69800	0.00000	0.00000
0.00000	0.00000	0.60000	0.00000	0.00000
0.00000	0.00000	0.68400	0.00000	0.00000
0.00000	0.00000	0.63250	0.00000	0.00000
0.00000	0.00000	0.68600	0.00000	0.00000
0.00000	0.00000	0.66750	0.00000	0.00000
0.00000	0.00000	0.70400	0.00000	0.00000
0.00000	0.00000	0.66500	0.00000	0.00000
0.00000	0.00000	0.72000	0.00000	0.00000
0.00000	0.00000	0.75250	0.00000	0.00000
0.00000	0.00000	0.77200	0.00000	0.00000
0.00000	0.00000	0.79500	0.00000	0.00000
0.00000	0.00000	0.78400	0.00000	0.00000
0.00000	0.00000	0.71750	0.00000	0.00000
0.00000	0.00000	0.74000	0.00000	0.00000
0.00000	0.00000	0.72750	0.00000	0.00000
0.00000	0.00000	0.72000	0.00000	0.00000
0.00000	0.00000	0.73250	0.00000	0.00000
0.00000	0.00000	0.73800	0.00000	0.00000
0.00000	0.00000	0.65500	0.00000	0.00000
0.00000	0.00000	0.68800	0.00000	0.00000
0.00000	0.00000	0.67000	0.00000	0.00000
0.00000	0.00000	0.71000	0.00000	0.00000
0.00000	0.00000	0.53250	0.00000	0.00000
0.00000	0.00000	0.63800	0.00000	0.00000
0.00000	0.00000	0.60500	0.00000	0.00000
0.00000	0.00000	0.66200	0.00000	0.00000
0.00000	0.00000	0.60500	0.00000	0.00000
0.00000	0.00000	0.65800	0.00000	0.00000
0.00000	0.00000	0.63750	0.00000	0.00000
0.00000	0.00000	0.67600	0.00000	0.00000
0.00000	0.00000	0.63250	0.00000	0.00000
0.00000	0.00000	0.65200	0.00000	0.00000
0.00000	0.00000	0.62500	0.00000	0.00000
0.00000	0.00000	0.67400	0.00000	0.00000
0.00000	0.00000	0.60000	0.00000	0.00000
0.00000	0.00000	0.66200	0.00000	0.00000
0.00000	0.00000	0.66750	0.00000	0.00000







0.00000	0.00000	0.00000	0.69000	0.00000
0.00000	0.00000	0.00000	0.66750	0.00000
0.00000	0.00000	0.00000	0.70800	0.00000
0.00000	0.00000	0.00000	0.62000	0.00000
0.00000	0.00000	0.00000	0.73000	0.00000
0.00000	0.00000	0.00000	0.73250	0.00000
0.00000	0.00000	0.00000	0.75000	0.00000
0.00000	0.00000	0.00000	0.73250	0.00000
0.00000	0.00000	0.00000	0.79000	0.00000
0.00000	0.00000	0.00000	0.70750	0.00000
0.00000	0.00000	0.00000	0.75000	0.00000
0.00000	0.00000	0.00000	0.64500	0.00000
0.00000	0.00000	0.00000	0.73400	0.00000
0.00000	0.00000	0.00000	0.57250	0.00000
0.00000	0.00000	0.00000	0.71400	0.00000
0.00000	0.00000	0.00000	0.68500	0.00000
0.00000	0.00000	0.00000	0.69400	0.00000
0.00000	0.00000	0.00000	0.71500	0.00000
0.00000	0.00000	0.00000	0.76400	0.00000
0.00000	0.00000	0.00000	0.78000	0.00000
0.00000	0.00000	0.00000	0.79800	0.00000
0.00000	0.00000	0.00000	0.74000	0.00000
0.00000	0.00000	0.00000	0.76400	0.00000
0.00000	0.00000	0.00000	0.78000	0.00000
0.00000	0.00000	0.00000	0.79000	0.00000
0.00000	0.00000	0.00000	0.00000	0.68875
0.00000	0.00000	0.00000	0.00000	0.45780
0.00000	0.00000	0.00000	0.00000	0.32263
0.00000	0.00000	0.00000	0.00000	0.24400
0.00000	0.00000	0.00000	0.00000	0.46000
0.00000	0.00000	0.00000	0.00000	0.35580
0.00000	0.00000	0.00000	0.00000	0.59250
0.00000	0.00000	0.00000	0.00000	0.37040
0.00000	0.00000	0.00000	0.00000	0.31875
0.00000	0.00000	0.00000	0.00000	0.25400
0.00000	0.00000	0.00000	0.00000	0.54000
0.00000	0.00000	0.00000	0.00000	0.36500
0.00000	0.00000	0.00000	0.00000	0.45375
0.00000	0.00000	0.00000	0.00000	0.26560
0.00000	0.00000	0.00000	0.00000	0.66000
0.00000	0.00000	0.00000	0.00000	0.51140
0.00000	0.00000	0.00000	0.00000	0.54125
0.00000	0.00000	0.00000	0.00000	0.45720
0.00000	0.00000	0.00000	0.00000	0.45250
0.00000	0.00000	0.00000	0.00000	0.32500
0.00000	0.00000	0.00000	0.00000	0.51125
0.00000	0.00000	0.00000	0.00000	0.43140
0.00000	0.00000	0.00000	0.00000	0.54125
0.00000	0.00000	0.00000	0.00000	0.20800
0.00000	0.00000	0.00000	0.00000	0.53250
0.00000	0.00000	0.00000	0.00000	0.28000
0.00000	0.00000	0.00000	0.00000	0.60750
0.00000	0.00000	0.00000	0.00000	0.33740
0.00000	0.00000	0.00000	0.00000	0.68250
0.00000	0.00000	0.00000	0.00000	0.46000
0.00000	0.00000	0.00000	0.00000	0.40375
0.00000	0.00000	0.00000	0.00000	0.41400
0.00000	0.00000	0.00000	0.00000	0.31375
0.00000	0.00000	0.00000	0.00000	0.52760
0.00000	0.00000	0.00000	0.00000	0.29125

0.00000	0.00000	0.00000	0.00000	0.31800
0.00000	0.00000	0.00000	0.00000	0.36625
0.00000	0.00000	0.00000	0.00000	0.29600
0.00000	0.00000	0.00000	0.00000	0.69500
0.00000	0.00000	0.00000	0.00000	0.32600
0.00000	0.00000	0.00000	0.00000	0.66875
0.00000	0.00000	0.00000	0.00000	0.48800
0.00000	0.00000	0.00000	0.00000	0.52500
0.00000	0.00000	0.00000	0.00000	0.46760
0.00000	0.00000	0.00000	0.00000	0.61000
0.00000	0.00000	0.00000	0.00000	0.33900
0.00000	0.00000	0.00000	0.00000	0.39000
0.00000	0.00000	0.00000	0.00000	0.47200
0.00000	0.00000	0.00000	0.00000	0.61125
0.00000	0.00000	0.00000	0.00000	0.25400
0.00000	0.00000	0.00000	0.00000	0.36250
0.00000	0.00000	0.00000	0.00000	0.38360
0.00000	0.00000	0.00000	0.00000	0.67875
0.00000	0.00000	0.00000	0.00000	0.52400
0.00000	0.00000	0.00000	0.00000	0.66875
0.00000	0.00000	0.00000	0.00000	0.25880
0.00000	0.00000	0.00000	0.00000	0.54625
0.00000	0.00000	0.00000	0.00000	0.41760
0.00000	0.00000	0.00000	0.00000	0.44750
0.00000	0.00000	0.00000	0.00000	0.33760
0.00000	0.00000	0.00000	0.00000	0.45375
0.00000	0.00000	0.00000	0.00000	0.33000
0.00000	0.00000	0.00000	0.00000	0.31750
0.00000	0.00000	0.00000	0.00000	0.17500
0.00000	0.00000	0.00000	0.00000	0.30375
0.00000	0.00000	0.00000	0.00000	0.18600
0.00000	0.00000	0.00000	0.00000	0.68500
0.00000	0.00000	0.00000	0.00000	0.26600
0.00000	0.00000	0.00000	0.00000	0.69750
0.00000	0.00000	0.00000	0.00000	0.53500
0.00000	0.00000	0.00000	0.00000	0.37000
0.00000	0.00000	0.00000	0.00000	0.55780
0.00000	0.00000	0.00000	0.00000	0.18750
0.00000	0.00000	0.00000	0.00000	0.40140
0.00000	0.00000	0.00000	0.00000	0.37250
0.00000	0.00000	0.00000	0.00000	0.31340
0.00000	0.00000	0.00000	0.00000	0.37500
0.00000	0.00000	0.00000	0.00000	0.29120
0.00000	0.00000	0.00000	0.00000	0.61375
0.00000	0.00000	0.00000	0.00000	0.41420
0.00000	0.00000	0.00000	0.00000	0.66875
0.00000	0.00000	0.00000	0.00000	0.34700
0.00000	0.00000	0.00000	0.00000	0.55750
0.00000	0.00000	0.00000	0.00000	0.52500
0.00000	0.00000	0.00000	0.00000	0.53750
0.00000	0.00000	0.00000	0.00000	0.37000
0.00000	0.00000	0.00000	0.00000	0.44375
0.00000	0.00000	0.00000	0.00000	0.39560
0.00000	0.00000	0.00000	0.00000	0.52125
0.00000	0.00000	0.00000	0.00000	0.35200
0.00000	0.00000	0.00000	0.00000	0.44500
0.00000	0.00000	0.00000	0.00000	0.32360
0.00000	0.00000	0.00000	0.00000	0.31625
0.00000	0.00000	0.00000	0.00000	0.32600
0.00000	0.00000	0.00000	0.00000	0.60375

0.00000	0.00000	0.00000	0.00000	0.40200
0.00000	0.00000	0.00000	0.00000	0.43250
0.00000	0.00000	0.00000	0.00000	0.37820
0.00000	0.00000	0.00000	0.00000	0.38000
0.00000	0.00000	0.00000	0.00000	0.39260
0.00000	0.00000	0.00000	0.00000	0.37375
0.00000	0.00000	0.00000	0.00000	0.48000
0.00000	0.00000	0.00000	0.00000	0.32250
0.00000	0.00000	0.00000	0.00000	0.29460
0.00000	0.00000	0.00000	0.00000	0.67875
0.00000	0.00000	0.00000	0.00000	0.51140
0.00000	0.00000	0.00000	0.00000	0.39125
0.00000	0.00000	0.00000	0.00000	0.39100
0.00000	0.00000	0.00000	0.00000	0.28250
0.00000	0.00000	0.00000	0.00000	0.38000
0.00000	0.00000	0.00000	0.00000	0.51250
0.00000	0.00000	0.00000	0.00000	0.41700
0.00000	0.00000	0.00000	0.00000	0.34125
0.00000	0.00000	0.00000	0.00000	0.35680
0.00000	0.00000	0.00000	0.00000	0.38000
0.00000	0.00000	0.00000	0.00000	0.34320
0.00000	0.00000	0.00000	0.00000	0.66000
0.00000	0.00000	0.00000	0.00000	0.32820
0.00000	0.00000	0.00000	0.00000	0.47500
0.00000	0.00000	0.00000	0.00000	0.28400
0.00000	0.00000	0.00000	0.00000	0.35750
0.00000	0.00000	0.00000	0.00000	0.49220
0.00000	0.00000	0.00000	0.00000	0.31625
0.00000	0.00000	0.00000	0.00000	0.42960
0.00000	0.00000	0.00000	0.00000	0.33375
0.00000	0.00000	0.00000	0.00000	0.27580
0.00000	0.00000	0.00000	0.00000	0.61750
0.00000	0.00000	0.00000	0.00000	0.31120
0.00000	0.00000	0.00000	0.00000	0.65750
0.00000	0.00000	0.00000	0.00000	0.51220
0.00000	0.00000	0.00000	0.00000	0.45125
0.00000	0.00000	0.00000	0.00000	0.39600
0.00000	0.00000	0.00000	0.00000	0.54125
0.00000	0.00000	0.00000	0.00000	0.44480
0.00000	0.00000	0.00000	0.00000	0.59875
0.00000	0.00000	0.00000	0.00000	0.30000
0.00000	0.00000	0.00000	0.00000	0.55375
0.00000	0.00000	0.00000	0.00000	0.48680
0.00000	0.00000	0.00000	0.00000	0.62250
0.00000	0.00000	0.00000	0.00000	0.42800
0.00000	0.00000	0.00000	0.00000	0.67375
0.00000	0.00000	0.00000	0.00000	0.35340
0.00000	0.00000	0.00000	0.00000	0.59750
0.00000	0.00000	0.00000	0.00000	0.45820
0.00000	0.00000	0.00000	0.00000	0.54125
0.00000	0.00000	0.00000	0.00000	0.47680
0.00000	0.00000	0.00000	0.00000	0.52500
0.00000	0.00000	0.00000	0.00000	0.57600
0.00000	0.00000	0.00000	0.00000	0.46750
0.00000	0.00000	0.00000	0.00000	0.50800
0.00000	0.00000	0.00000	0.00000	0.31000
0.00000	0.00000	0.00000	0.00000	0.40680
0.00000	0.00000	0.00000	0.00000	0.24750
0.00000	0.00000	0.00000	0.00000	0.33480
0.00000	0.00000	0.00000	0.00000	0.37500

0.00000	0.00000	0.00000	0.00000	0.30000
0.00000	0.00000	0.00000	0.00000	0.66000
0.00000	0.00000	0.00000	0.00000	0.16840
0.00000	0.00000	0.00000	0.00000	0.24500
0.00000	0.00000	0.00000	0.00000	0.22600
0.00000	0.00000	0.00000	0.00000	0.59750
0.00000	0.00000	0.00000	0.00000	0.23500
0.00000	0.00000	0.00000	0.00000	0.66750
0.00000	0.00000	0.00000	0.00000	0.51800
0.00000	0.00000	0.00000	0.00000	0.46500
0.00000	0.00000	0.00000	0.00000	0.23500
0.00000	0.00000	0.00000	0.00000	0.45000
0.00000	0.00000	0.00000	0.00000	0.46760
0.00000	0.00000	0.00000	0.00000	0.36250
0.00000	0.00000	0.00000	0.00000	0.39120
0.00000	0.00000	0.00000	0.00000	0.29500
0.00000	0.00000	0.00000	0.00000	0.30600
0.00000	0.00000	0.00000	0.00000	0.24000
0.00000	0.00000	0.00000	0.00000	0.18720
0.00000	0.00000	0.00000	0.00000	0.61125
0.00000	0.00000	0.00000	0.00000	0.15200
0.00000	0.00000	0.00000	0.00000	0.67875
0.00000	0.00000	0.00000	0.00000	0.11340
0.00000	0.00000	0.00000	0.00000	0.37500
0.00000	0.00000	0.00000	0.00000	0.24640
0.00000	0.00000	0.00000	0.00000	0.61500
0.00000	0.00000	0.00000	0.00000	0.41860
0.00000	0.00000	0.00000	0.00000	0.29375
0.00000	0.00000	0.00000	0.00000	0.26280
0.00000	0.00000	0.00000	0.00000	0.33125
0.00000	0.00000	0.00000	0.00000	0.45800
0.00000	0.00000	0.00000	0.00000	0.39125
0.00000	0.00000	0.00000	0.00000	0.27140
0.00000	0.00000	0.00000	0.00000	0.52375
0.00000	0.00000	0.00000	0.00000	0.20800
0.00000	0.00000	0.00000	0.00000	0.45375
0.00000	0.00000	0.00000	0.00000	0.25160
0.00000	0.00000	0.00000	0.00000	0.24750
0.00000	0.00000	0.00000	0.00000	0.30000
0.00000	0.00000	0.00000	0.00000	0.59625
0.00000	0.00000	0.00000	0.00000	0.40540
0.00000	0.00000	0.00000	0.00000	0.67125
0.00000	0.00000	0.00000	0.00000	0.34000
0.00000	0.00000	0.00000	0.00000	0.46625
0.00000	0.00000	0.00000	0.00000	0.51800
0.00000	0.00000	0.00000	0.00000	0.54750
0.00000	0.00000	0.00000	0.00000	0.48720
0.00000	0.00000	0.00000	0.00000	0.61000
0.00000	0.00000	0.00000	0.00000	0.33340
0.00000	0.00000	0.00000	0.00000	0.40000
0.00000	0.00000	0.00000	0.00000	0.38940
0.00000	0.00000	0.00000	0.00000	0.52500
0.00000	0.00000	0.00000	0.00000	0.38360
0.00000	0.00000	0.00000	0.00000	0.33125
0.00000	0.00000	0.00000	0.00000	0.47800
0.00000	0.00000	0.00000	0.00000	0.37250
0.00000	0.00000	0.00000	0.00000	0.54400
0.00000	0.00000	0.00000	0.00000	0.31750
0.00000	0.00000	0.00000	0.00000	0.52400
0.00000	0.00000	0.00000	0.00000	0.60000

0.00000	0.00000	0.00000	0.00000	0.43540
0.00000	0.00000	0.00000	0.00000	0.65625
0.00000	0.00000	0.00000	0.00000	0.36040
0.00000	0.00000	0.00000	0.00000	0.53000
0.00000	0.00000	0.00000	0.00000	0.33320
0.00000	0.00000	0.00000	0.00000	0.58875
0.00000	0.00000	0.00000	0.00000	0.39120
0.00000	0.00000	0.00000	0.00000	0.36625
0.00000	0.00000	0.00000	0.00000	0.49160
0.00000	0.00000	0.00000	0.00000	0.25125
0.00000	0.00000	0.00000	0.00000	0.55600
0.00000	0.00000	0.00000	0.00000	0.67000
0.00000	0.00000	0.00000	0.00000	0.36760
0.00000	0.00000	0.00000	0.00000	0.15000
0.00000	0.00000	0.00000	0.00000	0.34460
0.00000	0.00000	0.00000	0.00000	0.28500
0.00000	0.00000	0.00000	0.00000	0.35060
0.00000	0.00000	0.00000	0.00000	0.61125
0.00000	0.00000	0.00000	0.00000	0.25380
0.00000	0.00000	0.00000	0.00000	0.44875
0.00000	0.00000	0.00000	0.00000	0.55440
0.00000	0.00000	0.00000	0.00000	0.46250
0.00000	0.00000	0.00000	0.00000	0.51580
0.00000	0.00000	0.00000	0.00000	0.54875
0.00000	0.00000	0.00000	0.00000	0.11640
0.00000	0.00000	0.00000	0.00000	0.67125
0.00000	0.00000	0.00000	0.00000	0.44400
0.00000	0.00000	0.00000	0.00000	0.62250
0.00000	0.00000	0.00000	0.00000	0.44000
0.00000	0.00000	0.00000	0.00000	0.65500
0.00000	0.00000	0.00000	0.00000	0.37420
0.00000	0.00000	0.00000	0.00000	0.66625
0.00000	0.00000	0.00000	0.00000	0.43040
0.00000	0.00000	0.00000	0.00000	0.60000
0.00000	0.00000	0.00000	0.00000	0.41940
0.00000	0.00000	0.00000	0.00000	0.61000
0.00000	0.00000	0.00000	0.00000	0.51900
0.00000	0.00000	0.00000	0.00000	0.09625
0.00000	0.00000	0.00000	0.00000	0.41720
0.00000	0.00000	0.00000	0.00000	0.10750
0.00000	0.00000	0.00000	0.00000	0.48800
0.00000	0.00000	0.00000	0.00000	0.23250
0.00000	0.00000	0.00000	0.00000	0.51040
0.00000	0.00000	0.00000	0.00000	0.53500
0.00000	0.00000	0.00000	0.00000	0.44940
0.00000	0.00000	0.00000	0.00000	0.07250
0.00000	0.00000	0.00000	0.00000	0.14080
0.00000	0.00000	0.00000	0.00000	0.31000
0.00000	0.00000	0.00000	0.00000	0.37640
0.00000	0.00000	0.00000	0.00000	0.18125
0.00000	0.00000	0.00000	0.00000	0.45920
0.00000	0.00000	0.00000	0.00000	0.46750
0.00000	0.00000	0.00000	0.00000	0.01700
0.00000	0.00000	0.00000	0.00000	0.38250
0.00000	0.00000	0.00000	0.00000	0.02000
0.00000	0.00000	0.00000	0.00000	0.30875
0.00000	0.00000	0.00000	0.00000	0.44340
0.00000	0.00000	0.00000	0.00000	0.38500
0.00000	0.00000	0.00000	0.00000	0.18540
0.00000	0.00000	0.00000	0.00000	0.39625

0.00000	0.00000	0.00000	0.00000	0.26140
0.00000	0.00000	0.00000	0.00000	0.66750
0.00000	0.00000	0.00000	0.00000	0.33500
0.00000	0.00000	0.00000	0.00000	0.37500
0.00000	0.00000	0.00000	0.00000	0.50040
0.00000	0.00000	0.00000	0.00000	0.62125
0.00000	0.00000	0.00000	0.00000	0.49720
0.00000	0.00000	0.00000	0.00000	0.45125
0.00000	0.00000	0.00000	0.00000	0.18600
0.00000	0.00000	0.00000	0.00000	0.43625
0.00000	0.00000	0.00000	0.00000	0.14440
0.00000	0.00000	0.00000	0.00000	0.53000
0.00000	0.00000	0.00000	0.00000	0.16280
0.00000	0.00000	0.00000	0.00000	0.59125
0.00000	0.00000	0.00000	0.00000	0.28880
0.00000	0.00000	0.00000	0.00000	0.61625
0.00000	0.00000	0.00000	0.00000	0.44320
0.00000	0.00000	0.00000	0.00000	0.09250
0.00000	0.00000	0.00000	0.00000	0.25480
0.00000	0.00000	0.00000	0.00000	0.05875
0.00000	0.00000	0.00000	0.00000	0.36800
0.00000	0.00000	0.00000	0.00000	0.16000
0.00000	0.00000	0.00000	0.00000	0.24140
0.00000	0.00000	0.00000	0.00000	0.15125
0.00000	0.00000	0.00000	0.00000	0.25600
0.00000	0.00000	0.00000	0.00000	0.08875
0.00000	0.00000	0.00000	0.00000	0.32120
0.00000	0.00000	0.00000	0.00000	0.24375
0.00000	0.00000	0.00000	0.00000	0.40740
0.00000	0.00000	0.00000	0.00000	0.52875
0.00000	0.00000	0.00000	0.00000	0.43380
0.00000	0.00000	0.00000	0.00000	0.66750
0.00000	0.00000	0.00000	0.00000	0.03580
0.00000	0.00000	0.00000	0.00000	0.22250
0.00000	0.00000	0.00000	0.00000	0.33920
0.00000	0.00000	0.00000	0.00000	0.32750
0.00000	0.00000	0.00000	0.00000	0.21440
0.00000	0.00000	0.00000	0.00000	0.46375
0.00000	0.00000	0.00000	0.00000	0.30600
0.00000	0.00000	0.00000	0.00000	0.46500
0.00000	0.00000	0.00000	0.00000	0.49200
0.00000	0.00000	0.00000	0.00000	0.32000
0.00000	0.00000	0.00000	0.00000	0.20020
0.00000	0.00000	0.00000	0.00000	0.32250
0.00000	0.00000	0.00000	0.00000	0.07660
0.00000	0.00000	0.00000	0.00000	0.37875
0.00000	0.00000	0.00000	0.00000	0.41460
0.00000	0.00000	0.00000	0.00000	0.03125
0.00000	0.00000	0.00000	0.00000	0.15400
0.00000	0.00000	0.00000	0.00000	0.44000
0.00000	0.00000	0.00000	0.00000	0.13720
0.00000	0.00000	0.00000	0.00000	0.50875
0.00000	0.00000	0.00000	0.00000	0.26840
0.00000	0.00000	0.00000	0.00000	0.59375
0.00000	0.00000	0.00000	0.00000	0.11400
0.00000	0.00000	0.00000	0.00000	0.52000
0.00000	0.00000	0.00000	0.00000	0.41960
0.00000	0.00000	0.00000	0.00000	0.61000
0.00000	0.00000	0.00000	0.00000	0.54400
0.00000	0.00000	0.00000	0.00000	0.69000

0.00000	0.00000	0.00000	0.00000	0.44420
0.00000	0.00000	0.00000	0.00000	0.47250
0.00000	0.00000	0.00000	0.00000	0.40440
0.00000	0.00000	0.00000	0.00000	0.53750
0.00000	0.00000	0.00000	0.00000	0.34540
0.00000	0.00000	0.00000	0.00000	0.36625
0.00000	0.00000	0.00000	0.00000	0.43400
0.00000	0.00000	0.00000	0.00000	0.30375
0.00000	0.00000	0.00000	0.00000	0.34640
0.00000	0.00000	0.00000	0.00000	0.32500
0.00000	0.00000	0.00000	0.00000	0.35840
0.00000	0.00000	0.00000	0.00000	0.46500
0.00000	0.00000	0.00000	0.00000	0.41160
0.00000	0.00000	0.00000	0.00000	0.45750
0.00000	0.00000	0.00000	0.00000	0.39500
0.00000	0.00000	0.00000	0.00000	0.53750
0.00000	0.00000	0.00000	0.00000	0.30360
0.00000	0.00000	0.00000	0.00000	0.38625
0.00000	0.00000	0.00000	0.00000	0.53400
0.00000	0.00000	0.00000	0.00000	0.60375
0.00000	0.00000	0.00000	0.00000	0.36200
0.00000	0.00000	0.00000	0.00000	0.39375
0.00000	0.00000	0.00000	0.00000	0.43300
0.00000	0.00000	0.00000	0.00000	0.44625
0.00000	0.00000	0.00000	0.00000	0.33080
0.00000	0.00000	0.00000	0.00000	0.53250
0.00000	0.00000	0.00000	0.00000	0.25500
0.00000	0.00000	0.00000	0.00000	0.66875
0.00000	0.00000	0.00000	0.00000	0.49000
0.00000	0.00000	0.00000	0.00000	0.41125
0.00000	0.00000	0.00000	0.00000	0.40000
0.00000	0.00000	0.00000	0.00000	0.59875
0.00000	0.00000	0.00000	0.00000	0.32600
0.00000	0.00000	0.00000	0.00000	0.38500
0.00000	0.00000	0.00000	0.00000	0.42000
0.00000	0.00000	0.00000	0.00000	0.59125
0.00000	0.00000	0.00000	0.00000	0.31380
0.00000	0.00000	0.00000	0.00000	0.46250
0.00000	0.00000	0.00000	0.00000	0.52700
0.00000	0.00000	0.00000	0.00000	0.66250
0.00000	0.00000	0.00000	0.00000	0.39040
0.00000	0.00000	0.00000	0.00000	0.59625
0.00000	0.00000	0.00000	0.00000	0.39900
0.00000	0.00000	0.00000	0.00000	0.46000
0.00000	0.00000	0.00000	0.00000	0.43700
0.00000	0.00000	0.00000	0.00000	0.55125
0.00000	0.00000	0.00000	0.00000	0.40040
0.00000	0.00000	0.00000	0.00000	0.59625
0.00000	0.00000	0.00000	0.00000	0.38240
0.00000	0.00000	0.00000	0.00000	0.38625
0.00000	0.00000	0.00000	0.00000	0.56080
0.00000	0.00000	0.00000	0.00000	0.47125
0.00000	0.00000	0.00000	0.00000	0.47020
0.00000	0.00000	0.00000	0.00000	0.70625
0.00000	0.00000	0.00000	0.00000	0.38680
0.00000	0.00000	0.00000	0.00000	0.68250
0.00000	0.00000	0.00000	0.00000	0.52140
0.00000	0.00000	0.00000	0.00000	0.46000
0.00000	0.00000	0.00000	0.00000	0.49740
0.00000	0.00000	0.00000	0.00000	0.70250

0.00000	0.00000	0.00000	0.00000	0.41060
0.00000	0.00000	0.00000	0.00000	0.40375
0.00000	0.00000	0.00000	0.00000	0.40200
0.00000	0.00000	0.00000	0.00000	0.70375
0.00000	0.00000	0.00000	0.00000	0.38320

#Pattern 1 (Output)

[ 381 1 ]

0.6928114  
0.4841138  
0.7859501  
0.7442806  
0.6656302  
0.5357571  
0.7852719  
0.7840075  
0.5312730  
0.7522102  
0.6217963  
0.6593780  
0.7443864  
0.6215659  
0.7007009  
0.6576208  
0.6984804  
0.7280053  
0.6607509  
0.5770292  
0.7459660  
0.7210989  
0.6737091  
0.6022329  
0.6963605  
0.7627404  
0.6964363  
0.4688572  
0.4951552  
0.4495454  
0.7319104  
0.5743988  
0.4776929  
0.4543184  
0.7514892  
0.6777464  
0.5810397  
0.5026254  
0.5369469  
0.4869223  
0.7431614  
0.6151760  
0.7526598  
0.7099249  
0.7232994  
0.5852725  
0.6511165  
0.7927537  
0.6588797  
0.7221526



0.5099433  
0.6384129  
0.7855125  
0.7456983  
0.7491940  
0.7317743  
0.7850291  
0.7085567  
0.7849268  
0.5966785  
0.7134644  
0.7301459  
0.6796501  
0.7800586  
0.6419890  
0.5604618  
0.7378000  
0.7294400  
0.7288400  
0.7280800  
0.7055600  
0.6192800  
0.5608000  
0.5596000  
0.5427200  
0.5274400  
0.5166000  
0.5136800  
0.5128800  
0.5089200  
0.4451200  
0.4358400  
0.4272400  
0.4118400  
0.3926800  
0.3865600  
0.3817600  
0.7728400  
0.7784800  
0.7733600  
0.7754400  
0.7775600  
0.7850000  
0.7852000  
0.7976800  
0.7992800  
0.7925600  
0.7912400  
0.7946000  
0.7829200  
0.7853600  
0.7582400  
0.7717600  
0.7703600  
0.7815200  
0.7774800  
0.7775200  
0.7732800  
0.7868800  
0.7914800

0.7826400  
0.7793200  
0.7913600  
0.7973600  
0.8000800  
0.7900800  
0.7917600  
0.7993200  
0.7857200  
0.7851200  
0.7853200  
0.7900800  
0.7954800  
0.7910800  
0.7919200  
0.7964400  
0.7974400  
0.7870400  
0.7910400  
0.7870400  
0.7994800  
0.7885200  
0.7890000  
0.7890800  
0.7921200  
0.7863600  
0.7908000  
0.7932400  
0.7854000  
0.7648000  
0.7814400  
0.7806400  
0.7890800  
0.7879200  
0.7889600  
0.7852400  
0.7823600  
0.7872000  
0.7998000  
0.7980800  
0.7954800  
0.7915600  
0.7936400  
0.7961600  
0.7938000  
0.7938000  
0.7891600  
0.7803200  
0.7848800  
0.7789200  
0.7955600  
0.7992800  
0.7982800  
0.8008000  
0.8014000  
0.7883200  
0.7804800  
0.7713600  
0.7842000  
0.7858400

0.7832400  
0.7949600  
0.7958000  
0.7926000  
0.7868000  
0.7806400  
0.7878400  
0.7948800  
0.7976800  
0.7960400  
0.7970400  
0.7707298  
0.5502799  
0.6591733  
0.7294571  
0.5238049  
0.7033380  
0.6471265  
0.7684482  
0.7224287  
0.5952966  
0.7098724  
0.5572064  
0.6877577  
0.7043674  
0.7661077  
0.6765609  
0.7698029  
0.5894997  
0.5555114  
0.6520264  
0.7762669  
0.7562637  
0.6946319  
0.7675968  
0.5949616  
0.7443250  
0.7789341  
0.6446660  
0.7408185  
0.6545484  
0.6632845  
0.5383565  
0.5478800  
0.5402254  
0.7781351  
0.7795954  
0.6336534  
0.5980383  
0.4560895  
0.7427006  
0.6097993  
0.7730997  
0.6655893  
0.7140675  
0.6453224  
0.6473420  
0.6369467  
0.7065337  
0.7392413

0.7321559  
0.7715922  
0.6131309  
0.7757824  
0.6733335  
0.6627782  
0.7339564  
0.6481274  
0.5848232  
0.6006744  
0.5245683  
0.7661430  
0.7382268  
0.5618087  
0.6309479  
0.7703933  
0.6364463  
0.7252262  
0.5571314  
0.7686427  
0.7198071  
0.6285403  
0.7342388  
0.7545908  
0.7776989  
0.7734290  
0.7396442  
0.7044266  
0.6477159  
0.5254847  
0.5019044  
0.6077387  
0.7724212  
0.4821226  
0.7512121  
0.7276182  
0.6991985  
0.6317520  
0.5226643  
0.4650155  
0.5031172  
0.7461173  
0.6817701  
0.7499723  
0.5833372  
0.4511847  
0.5090775  
0.5367304  
0.7054510  
0.6414536  
0.7679530  
0.7549688  
0.6614175  
0.6012653  
0.6857359  
0.7278030  
0.7645888  
0.7707149  
0.7433548  
0.6320436

0.5967466  
0.7125925  
0.7477235  
0.7748445  
0.6281126  
0.5171101  
0.5980857  
0.4355043  
0.7808194  
0.7744549  
0.2851411  
0.7441332  
0.7469242  
0.6775760  
0.6839847  
0.7318774  
0.7715892  
0.7523169  
0.7680739  
0.7725975  
0.7453906  
0.2896376  
0.7189819  
0.7247461  
0.1540857  
0.2462264  
0.6855403  
0.4942229  
0.5626994  
0.4728796  
0.7670852  
0.7723010  
0.4286227  
0.4201178  
0.4197848  
0.5919439  
0.7529250  
0.5939001  
0.7067870  
0.6503603  
0.6221641  
0.6997845  
0.7377080  
0.7402939  
0.2376994  
0.6916435  
0.3867572  
0.5122259  
0.7731702  
0.4191589  
0.3514570  
0.7678218  
0.5249806  
0.4911828  
0.6159573  
0.5183890  
0.7508177  
0.7755595  
0.7215664  
0.6879292

0.6403281  
0.7265701  
0.6630596  
0.6983990  
0.7393774  
0.7378917  
0.6384768  
0.7796723  
0.6731049  
0.7420250  
0.5884677  
0.5422730  
0.7663037  
0.7103273  
0.6225330  
0.7334422  
0.6434691  
0.7668550  
0.6779550  
0.6709214  
0.7273973  
0.6924223  
0.6617528  
0.7767911  
0.7466022  
0.6710398  
0.7696459  
0.7530422  
0.6762703  
0.6425257  
0.6814468