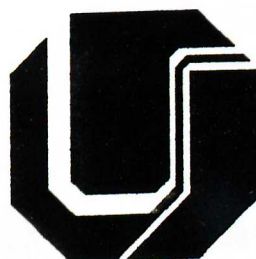


**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



**O USO DO TEOREMA DA FUNÇÃO IMPLÍCITA NO
DIMENSIONAMENTO DE REDES NEURAIAS
MULTI-CAMADAS**

MARCELO HENRIQUE BELONSI

FEVEREIRO

1996

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

O USO DO TEOREMA DA FUNÇÃO IMPLÍCITA NO
DIMENSIONAMENTO DE REDES NEURAIS
MULTI-CAMADAS

Dissertação apresentada por Marcelo Henrique Belonsi à
Universidade Federal de Uberlândia para obtenção do título
de Mestre em Engenharia Elétrica aprovada em 26/02/96 pela
Banca Examinadora:

Prof. Edilberto Pereira Teixeira, Dr. UFU (orientador)

Prof. Gilberto Arantes Carrijo, Dr. UFU

Prof. Weber Martins, Ph.D UFG

681.3:007.52 B452u /TES/F
DIRBI/UFU 00300/96



1000022550

127

**O USO DO TEOREMA DA FUNÇÃO IMPLÍCITA NO
DIMENSIONAMENTO DE REDES NEURAIS
MULTI-CAMADAS**

MARCELO HENRIQUE BELONSI

Dissertação apresentada por Marcelo Henrique Belonsi à Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Prof. Edilberto Pereira. Teixeira, Dr.
Orientador

Prof. Haroldo Rodrigues de Azevedo, Dr.
Coordenador do Curso de Pós Graduação

UNIVERSIDADE FEDERAL DE UBERLÂNDIA	
IBRBI - DIRETORIA DE BIBLIOTECAS	
Procedência _____	
Valor R\$ _____	
Valor R\$	Doação
Fac. em	196
C.F.D.	
N. Fiscal/Fol.	de
Req.	00300/96
Data	12.04.96
Vol.	Ex. 1

FU-00005609-3

Dedico esta dissertação a meus pais, PAULO e MARIA APARECIDA; irmãos, MARIA DO ROSÁRIO e PAULO ROBERTO; e namorada, JACQUELAINE pelo incentivo e dedicação.

AGRADECIMENTOS

A Deus, pois é ele a fonte de tudo;

Aos meus pais Paulo Francisco Belonsi e Maria Aparecida Bravo Belonsi por tudo que fizeram e representaram;

Ao orientador, Prof. Dr. Edilberto Pereira Teixeira;

A todos que de alguma forma contribuíram para a realização deste trabalho;

A CAPES pelo apoio financeiro.

RESUMO

Este trabalho apresenta um método para o dimensionamento de redes neurais artificiais de uma única camada intermediária, levando-se em consideração as condições de treinabilidade e a precisão da aproximação que tais redes devem proporcionar.

A precisão da aproximação funcional está intimamente ligada à estrutura de tais redes, portanto, para se garantir uma aproximação satisfatória, apresenta-se um método de se dimensionar a camada intermediária de tais tipos de redes, usando-se para isto o teorema da função implícita.

Simulações do método proposto, comprovam os resultados teóricos obtidos.

ABSTRACT

This work presents a method for dimensioning artificial neural networks with a single hidden layer, considering the trainability conditions and the approximation accuracy.

The functional approximation accuracy is highly dependent on the network structure. Therefore, to guarantee a reasonable approximation, a method is presented to determine the number of elements in the hidden layer, using the implicit function theorem.

Some simulations illustrates the theoretical results obtained in this work.

SUMÁRIO

CAPÍTULO 1	1
Introdução	1
	5
CAPÍTULO 2	5
2.1 Introdução	7
2.2 Definições	10
2.3 Redes neurais como aproximadores universais	10
2.4 Conclusão	27
	28
CAPÍTULO 3	28
3.1 Introdução	29
3.2 Arquitetura de redes neurais	32
3.3 Treinabilidade	42
3.4 Conclusão	
	44
CAPÍTULO 4	44
4.1 Introdução	46
4.2 Definições	47
4.3 Equações da rede neural	
4.4 O efeito da tolerância durante o treinamento da rede neural	63
4.5 Conclusão	65
	66
CAPÍTULO 5	66
5.1 Simulações	77
5.2 Conclusão	

REFERÊNCIAS BIBLIOGRÁFICAS

78

ANEXO

81

Tabelas

81

SIMBOLOGIA

\exists	- Existe.
\forall	- Para todo.
\Rightarrow	- Implica.
$[a,b]$	- Intervalo fechado de a até b.
$f_n \rightarrow f$	- f_n converge para f .
\mathbb{N}	- Conjunto dos números naturais.
\mathbb{R}	- Conjunto dos números reais.
RNA	- Rede Neural Artificial.
A_{tm}	- Conjunto de amostras de treinamento de uma rede neural.
\otimes	- Símbolo de operação lógica ou matemática.
\oplus	- Símbolo de operação lógica OU.
$\text{Det}[A]$	- Determinante da matriz quadrada A.

CAPÍTULO 1

INTRODUÇÃO

Desde o desenvolvimento do Perceptron (1950), houve inúmeras tentativas de se imitar o desempenho do cérebro humano no que tange à capacidade de aprendizagem até que, no ano de 1969, Minsky e Papert demonstraram que o perceptron de duas camadas é incapaz de representar ou aproximar quaisquer tipos de mapeamentos, mas, apenas, os linearmente separáveis. Isto ocasionou um grande esfriamento no desenvolvimento da teoria das redes neurais. Apesar das afirmações de Minsky e Papert, era publicamente difundida a possibilidade de redes neurais de várias camadas implementarem quaisquer mapeamentos. Somente nos últimos anos (década de 80 em diante), alguns pesquisadores obtiveram excelentes resultados utilizando redes neurais como aproximadores de mapeamentos de um espaço de dimensão finita para outro. Recentemente, esse tipo de pesquisa veio à tona com a enorme variedade de aplicações que esses dispositivos podem realizar. Dentre elas, destacam-se problemas de otimização, reconhecimento de imagem e voz, etc. Vários dispositivos com capacidade de aprender por treinamento repetitivo

foram desenvolvidos, mas com pouco grau de aproveitamento prático, em áreas que não sejam das ciências exatas.

Estes dispositivos foram denominados Redes Neurais Artificiais (RNAs), e podem ser classificados de duas formas distintas. A primeira classificação considera o método de treinamento considerando-se as redes de aprendizado supervisionado e não-supervisionado. A segunda classificação considera o número de camadas da rede. Neste trabalho concentrou-se apenas nas redes de aprendizado supervisionado de três camadas. Em Rumelhart [1] encontra-se uma discussão detalhada de tais tipos de redes.

Basicamente, as redes neurais artificiais são constituídas de vários elementos de processamento interconectados. O tipo apresentado na figura 3.1 tem como característica principal a possibilidade de aprender mapeamentos de estruturas de sistemas bastante complexos. As redes de três camadas constituem-se de uma camada intermediária que tem dimensão igual à dimensão dos padrões do conjunto de treinamento, uma camada intermediária, cuja dimensão é responsável pela treinabilidade e a precisão da aproximação e, finalmente, uma camada de saída que tem dimensão igual à saída desejada. Basicamente, o treinamento da rede é feito através do ajuste dos elementos da matriz de pesos (parâmetros de ajuste). Este treinamento pode ser feito por diversos tipos de algoritmos especificamente este trabalho, focalizou-se no algoritmo de propagação retroativa (algoritmo backpropagation) proposto por Rumelhart [1]. O treinamento da

rede, por esse tipo de algoritmo procede-se por duas etapas: a primeira chama-se de fase de treinamento, onde os padrões são apresentados à rede e fluem da entrada para a saída. Os pesos das conexões internas da rede são ajustados comparando-se a saída calculada com o valor desejado da saída e ajustando-se os pesos até que o erro torne-se mínimo. A segunda fase chama-se etapa de operação (também conhecida como fase de reconhecimento). Nesta etapa, os pesos não são mais alterados (embora que existam algoritmos onde o treinamento é feito on-line, ou seja, na etapa de operação os pesos podem ser alterados), sendo que o conjunto de padrões de operação é apresentado à rede e esta fornece os valores correspondente da saída.

Para se estimar o número de elementos na camada intermediária, deve-se considerar as condições de treinabilidade da rede e precisão desejada no mapeamento. Em geral, aumentando-se o número de elementos na camada intermediária obtém-se uma representação mais refinada. Existe, entretanto o risco de over-fitting.

O capítulo 2 deste trabalho, dedica-se à prova apresentada por Hornik [7] de que redes neurais com apenas uma camada intermediária são capazes de representar arbitrariamente qualquer tipo de mapeamento de um espaço de dimensão finita para outro.

O capítulo 3, são propostas e demonstradas algumas proposições em relação às condições de treinabilidade das redes neurais artificiais, onde se estabelece um número mínimo de parâmetros de ajuste (ou pesos) que uma RNA deve ter para que a mesma possa ser treinável.

O capítulo 4, apresenta-se um método prático para se estimar o número mínimo de elementos na camada intermediária, baseado na condições de treinabilidade da rede.

No capítulo 5, apresentam-se os resultados práticos, da verificação da análise teórica obtida no capítulo 4.

CAPÍTULO 2

O EMPREGO DE REDES NEURAIS EM APROXIMAÇÃO FUNCIONAL

2.1 INTRODUÇÃO

Nos últimos anos, diversos autores (Carrol & Dickinson, 1989; Cybenko, 1989; Funahashi, 1989, Hornik, 1991; Hornik, Stinchcombe, & White, 1989) começaram a explorar as redes neurais como aproximadores de mapeamentos de um espaço de dimensão finita para outro. Importantes conclusões foram atingidas, mostrando a aplicabilidade de redes como aproximadores universais. Considerando-se esses resultados, apresenta-se, neste capítulo, uma análise dos pontos relevantes, mostrando que redes neurais artificiais com apenas uma camada intermediária são capazes de aproximações precisas e satisfatórias.

Para isto, é necessário que se considere o fato de que toda função analítica admite em torno de cada ponto de seu domínio, um desenvolvimento de Taylor, o qual representa a função como soma de séries de potências: $f(x) = \sum_{i=0}^{\infty} b_i (x-a)^i$, onde b_i são constantes com $i = 1, 2, \dots, \infty$ e a é um ponto qualquer pertencente ao domínio de $f(x)$.

Seja

$$p_n(x) = b_0 + b_1(x-a) + b_2(x-a)^2 + \dots + b_n(x-a)^n$$

onde p_n é um polinômio e $f(x) = \lim_{n \rightarrow \infty} p_n(x)$, para todo x pertencente ao intervalo de convergência da série. Além disso, em cada ponto tem-se $p_n \rightarrow f$, para cada subconjunto compacto desse intervalo de convergência.

Um resultado demonstrado por K. Weierstrass em 1855, generaliza a situação acima descrita. Segundo Weierstrass, "*Qualquer função contínua f definida em $[a,b]$ pode ser aproximada por uma sucessão de polinômios que convergem uniformemente para f em $[a,b]$* ". Assim dada f contínua em $[a,b]$ e dado $\varepsilon > 0$, existe um polinômio p tal que $|f(x) - p(x)| < \varepsilon, \forall x \in [a,b]$.

Para mostrar o proposto no primeiro parágrafo aplica-se o teorema da aproximação de Weierstrass, bem como o teorema da superposição de Kolmogorov,

“Qualquer função contínua de várias variáveis $f(x_1, x_2, \dots, x_n)$ pode ser representada por funções contínuas de uma única variável”. Se a função de ativação de cada elemento da rede neural for uma função contínua de valor real ela também poderá ser aproximada por um polinômio. Estes conceitos serão fundamentados através de algumas definições e demonstrando-se o teorema de Weierstrass acima enunciado.

2.2 DEFINIÇÕES

Definição 2.2.1

Uma função h é dita de classe C^1 quando a primeira derivada de h existir e for contínua. No caso em que h tenha domínio em \mathbb{R}^n diz-se que h é de classe C^1 quando as primeiras derivadas parciais de h existirem e forem todas contínuas.

Definição 2.2.2

Uma sucessão de funções $f_n : X \rightarrow \mathbb{R}$, converge uniformemente para a função $f : X \rightarrow \mathbb{R}$ quando, $\varepsilon > 0$ dado, $\exists n_0 \in \mathbb{N}$ tal que $\forall n > n_0 \Rightarrow |f_n(x) - f(x)| < \varepsilon$, $\forall x \in X$.

Definição 2.2.3

Seja $\sigma : \mathbb{R} \rightarrow [0, 1]$ definida por:

$$\sigma(\lambda) = \frac{1}{1 + e^{-\lambda}} \quad (2.1)$$

onde $\lambda \in \mathbb{R}$. A esta função, definida desta maneira, denomina-se função sigmóide.

Observação:

A função sigmóide definida acima tem as seguintes propriedades:

Propriedade 2.2.1

$$(1) \left\{ \begin{array}{l} \lim_{\lambda \rightarrow -\infty} \sigma(\lambda) = 0 \\ e \\ \lim_{\lambda \rightarrow +\infty} \sigma(\lambda) = 1 \end{array} \right.$$

(2) $\sigma(\lambda)$ definida como em (2.1) é crescente;

$$(3) \frac{d\sigma(\lambda)}{d\lambda} = \sigma(\lambda)[1 - \sigma(\lambda)]$$

(4) A função sigmóide ($\sigma(\lambda)$) definida como em (2.1) é de classe C^1 .

A prova das propriedades anteriores é imediata.

2.3 REDES NEURAIS COMO APROXIMADORES UNIVERSAIS

Em seguida, são enunciados e demonstrados alguns lemas que são usados na prova do teorema da aproximação de Weierstrass.

Lema 2.3.1 [13]

Se $0 < \delta < 1$, então φ_n converge uniformemente para 0, $\forall |x| \geq \delta$, onde:

$$\varphi_n(x) = \begin{cases} 0, & |x| \geq 1 \\ \frac{(1-x^2)^n}{c_n}, & x \in [-1,1] \end{cases} \quad (2.2)$$

Prova:

Para cada $n \in \mathbb{N}$, seja:

$$C_n = \int_{-1}^{+1} (1-x^2)^n dx \quad (2.3)$$

Separando o integrando de (2.3) em um produto e ainda usando o fato de $x \in [0, 1]$ vem:

$$C_n = 2 \int_0^1 (1+x)^n (1-x)^n dx \quad (2.4)$$

De (2.4) e ainda $x \in [0, 1]$ vem:

$$C_n \geq 2 \int_0^1 (1-x)^n dx \quad (2.5)$$

Mas, por outro lado, segue-se (2.6), que verifica-se fazendo-se a mudança de variável $y = 1 - x$ e resolvendo-se a integral definida no intervalo $[0, 1]$.

$$2 \int_0^1 (1-x)^n dx = \frac{2}{n+1} \quad (2.6)$$

De (2.5) e (2.6), conclui-se:

$$C_n \geq \frac{2}{n+1} \quad (2.7)$$

Logo de (2.2) e (2.7) e ainda pela hipótese, segue-se que:

$$\delta \leq |x| \Rightarrow \varphi_n(x) \leq \frac{(1-\delta^2)^n \cdot (n+1)}{2} \quad (2.8)$$

Como

$$0 < 1 - \delta^2 < 1 \quad (2.9)$$

Vem

$$\lim_{n \rightarrow \infty} (1 - \delta^2)^n \cdot (n+1) = 0 \quad (2.10)$$

Portanto de (2.9) e (2.10) tem-se:

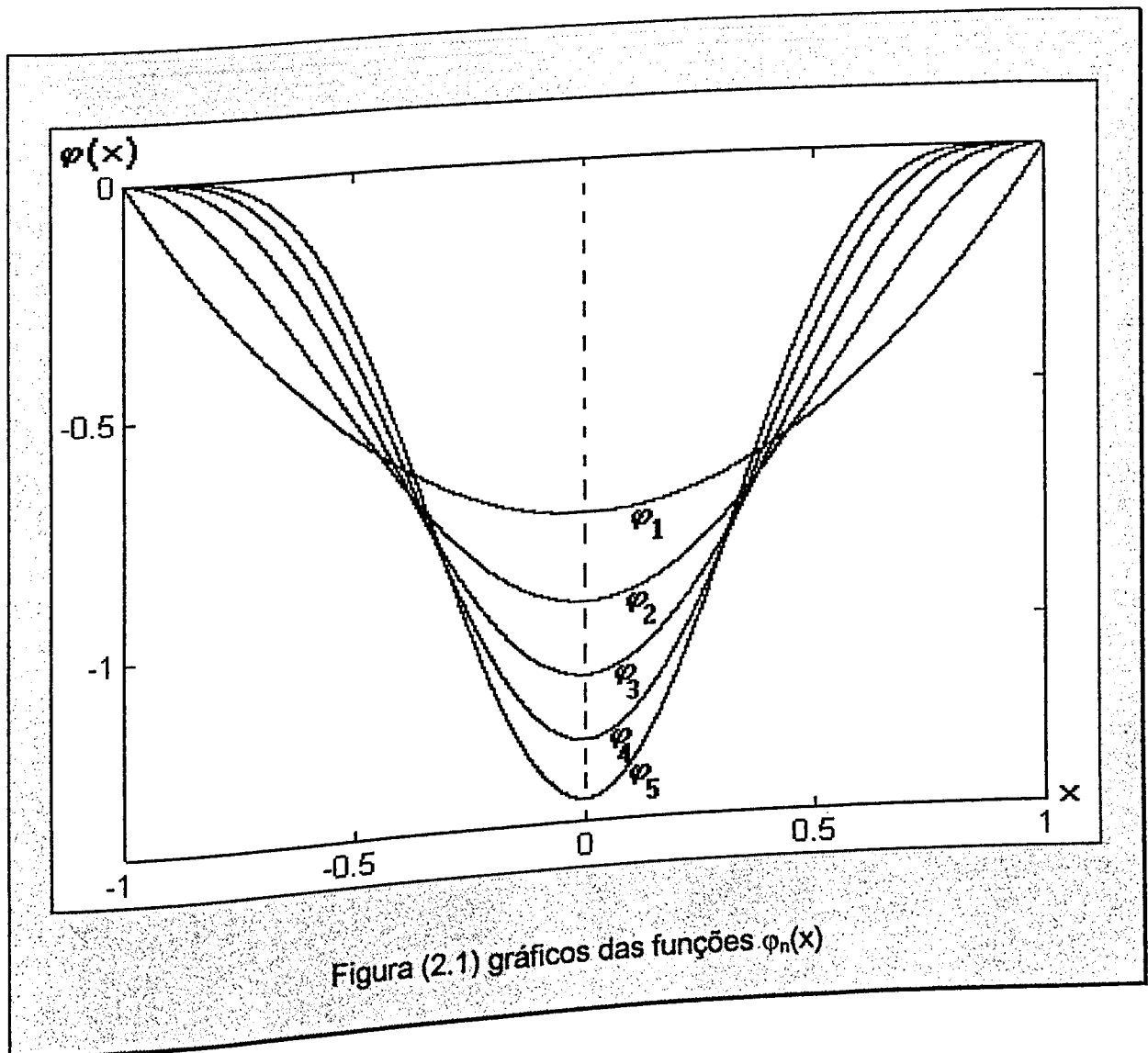
$$\lim_{n \rightarrow \infty} \varphi_n(x) = 0 \quad (2.11)$$

Logo da equação (2.1) conclui-se que φ_n converge uniformemente.

■

Observação:

A figura 2.1 abaixo mostra a forma aproximada dos gráficos das funções φ_n .



A área subtendida pelo eixo das abcissas e cada gráfico é igual a 1, o que obriga $\lim_{n \rightarrow \infty} \varphi_n(0) = \infty$. Se existisse $\lim_{n \rightarrow \infty} \varphi_n = \varphi$, ter-se-ia $\varphi(x) = 0, \forall x \neq 0$, e

ainda:

$$\int_{-\infty}^{+\infty} \varphi(x) dx = 1 \quad (2.12)$$

Lema 2.3.2 [13]

Seja $f: \mathbb{R} \rightarrow \mathbb{R}$ contínua, com $f(0) = f(1) = 0$. Pondo $f(x) = 0$ se $x \notin [0, 1]$,

$\forall x \in [0, 1]$ e $\forall n \in \mathbb{N}$, escreva:

$$p_n(x) = \int_{-1}^{+1} f(x+u)\varphi_n(u)du = \int_{-\infty}^{+\infty} f(x+u)\varphi_n(u)du \quad (2.13)$$

Então $p_n: [0, 1] \rightarrow \mathbb{R}$, assim definida é um polinômio.

Prova:

Da primeira igualdade de (2.13) e para $x \in [0, 1]$, fazendo $y = x + u$ tem-se:

$$p_n(x) = \int_{x-1}^{x+1} f(y) \varphi_n(y-x) dy \quad (2.14)$$

De (2.14) e ainda $f(x) = 0$ para $x \notin [0, 1]$, vem:

$$\int_{x-1}^{x+1} f(y) \varphi_n(y-x) dy = \int_0^1 f(y) \varphi_n(y-x) dy \quad (2.15)$$

De (2.14) e (2.15), segue-se:

$$p_n(x) = \int_0^1 f(y) \varphi_n(y-x) dy \quad (2.16)$$

Sendo $x, y \in [0, 1]$, temos $y - x \in [-1, 1]$, logo:

$$\varphi_n(y-x) = [1 - (y-x)^2]^n \quad (2.17)$$

De (2.17) e como $\varphi_n(y-x)$ é analítica (equação 2.2), vem:

$$[1 - (y - x)^2]^n = \sum_{i=0}^{2n} \xi_i(y) x^i \quad (2.18)$$

De (2.17) e (2.18), tem-se:

$$\varphi_n(y - x) = \sum_{i=0}^{2n} \xi_i(y) x^i \quad (2.19)$$

De (2.16) e (2.19), tem-se:

$$p_n(x) = \int_0^1 f(y) \sum_{i=0}^{2n} \xi_i(y) x^i dy \quad (2.19)$$

De (2.19) e usando a continuidade de $f(x)$, segue-se:

$$p_n(x) = \sum_{i=0}^{2n} x^i \int_0^1 f(y) \xi_i(y) dy \quad (2.20)$$

Pondo para cada $i = 0, 1, 2, \dots, 2n$

$$a_i = \int_0^1 f(y) \cdot \xi_i(y) dy \quad (2.21)$$

De (2.20) e (2.21) tem-se:

$$p_n(x) = \sum_{i=0}^{2n} a_i \cdot x^i \quad (2.22)$$

Logo (2.22) mostra que $p_n : [0,1] \rightarrow \mathbb{R}$, é um polinômio. ■

Lema 2.3.3 [13]

P_n converge uniformemente para f no intervalo $[0,1]$.

Prova:

Da equação (2.12) segue que:

$$\int_{-1}^{+1} \varphi_n(x) dx = 1 \quad (2.23)$$

De (2.23), temos:

$$f(x) = \int_{-1}^{+1} f(x) p_n(u) du \quad (2.24)$$

Logo, fazendo a diferença de (2.24) e (2.13), segue-se que, $\forall n \in \mathbb{N}$ e todo $x \in [0, 1]$, vale:

$$f(x) - p_n(x) = \int_{-1}^{+1} [f(x+u) - f(x)] \varphi_n(u) du \quad (2.25)$$

Como f é uniformemente contínua, dado qualquer $\varepsilon > 0$, $\exists \delta > 0$, tal que:

$$|u| < \delta \Rightarrow |f(x+u) - f(x)| < \frac{\varepsilon}{3} \quad (2.26)$$

Qualquer que seja $x \in [0, 1]$. Seja:

$$M = \sup_{x \in [0, 1]} |f(x)| \quad (2.27)$$

Pelo lema 2.3.1, $\exists n_0 \in \mathbb{N}$ tal que:

$$|u| \geq \delta \Rightarrow |\varphi_n(u)| < \frac{\varepsilon}{6M} \quad (2.28)$$

Logo, de (2.25) e $\forall n > n_0$, e todo $x \in [0, 1]$ tem-se:

$$|f(x) - p_n(x)| \leq A + B + C \quad (2.29)$$

Onde,

$$A = \int_{-1}^{-\delta} |f(x+u) - f(x)| \varphi_n(u) du < \frac{2M \cdot \varepsilon}{6M} = \frac{\varepsilon}{3} \quad (2.30)$$

$$B = \int_{-\delta}^{+\delta} |f(x+u) - f(x)| \varphi_n(u) du < \frac{\varepsilon}{3} \cdot \int_{-\delta}^{+\delta} \varphi_n(t) dt \leq \frac{\varepsilon}{3} \quad (2.31)$$

$$C = \int_{\delta}^1 |f(x+u) - f(x)| \varphi_n(u) du < \frac{2M \cdot \varepsilon}{6M} = \frac{\varepsilon}{3} \quad (2.32)$$

Logo de (2.29), (2.30), (2.31) e (2.32) temos:

$$|f(x) - p_n(x)| < \varepsilon \quad (2.33)$$

sempre que $n > n_0$.

A equação (2.33) mostra que $p_n(x)$ converge uniformemente para $f(x)$ sendo $x \in [0, 1]$. ■

Observação:

Os lemas acima mostram que toda função contínua $f : [0, 1] \rightarrow \mathbb{R}$ tal que $f(0) = f(1) = 0$ é limite uniforme de uma sequência de polinômios.

Teorema 2.3.1 (Weierstrass [13])

Dada uma função contínua $f : [a, b] \rightarrow \mathbb{R}$, existe uma sequência de polinômios (p_n) que converge uniformemente para f em $[a, b]$.

Prova:

Seja φ_n e c_n definida como nas equações (2.2) e (2.3), respectivamente. Logo φ_n é contínua e de Classe C' , e ainda tem-se:

$$\varphi_n(-x) = \varphi_n(x) \quad (2.34)$$

e ainda vemos que as equações (2.12) é igual à (2.23)

Pelo lema 2.3.1 φ_n converge uniformemente para zero

Pelo lema 2.3.2 existe p_n definida como na equação (2.13).

Pelo lema 2.3.3 p_n converge uniformemente para f .

Portanto existe uma sequência de polinômios p_n tal que:

$$|p_n(x) - f(x)| < \varepsilon$$

O que mostra que p_n converge uniformemente para f .

Teorema 2.3.2 (Kolmogorov [17])

Qualquer função contínua $f(x_1, x_2, \dots, x_n)$ de várias variáveis definidas em $I^n = [0, 1] \times [0, 1] \times \dots \times [0, 1]$ (produto cartesiano) pode ser representada na forma:

$$f(x_1, \dots, x_n) = \sum_{i=1}^{2n+1} g_i \left(\sum_{j=1}^n \psi_{ji}(x_j) \right) \quad (2.35)$$

Onde, g_i e ψ_{ji} são funções contínuas de uma variável. ■

Teorema 2.3.3 (Hornik [7])

Seja $f: I^n \rightarrow \mathbb{R}$, e $y(x) = \sum_{i=1}^n (W_i \sigma(P_i x + \alpha_i) + \beta_i)$ uma rede neural, com n elementos na camada intermediária. Então $y(x)$ converge uniformemente para f .

Prova:

Pelo teorema de Kolmogorov, equação (2.35) tem-se:

$$f(x_1, \dots, x_n) = \sum_{i=1}^{2n+1} g_i \left(\sum_{j=1}^n \psi_{ji}(x_j) \right) \quad (2.36)$$

Pelo teorema da aproximação de Weierstrass, deve existir polinômios p_n , q_n e r_n , tal que, $\forall \varepsilon' > 0$, segue-se:

$$A = \left| \sum_{i=1}^{2n+1} g_i \left(\sum_{j=1}^n \psi_{ji}(x_j) \right) - \sum_{i=1}^{2n+1} p_{n_i} \left(\sum_{j=1}^n q_{n_j}(x_j) \right) \right| < \frac{\varepsilon'}{2} \quad (2.37)$$

e

$$B = \left| \sum_{i=1}^n W_i \sigma(P_i x + \alpha_i) + \beta_i - \sum_{i=1}^n W_i r_n(P_i x + \alpha_i) + \beta_i \right| < \frac{\varepsilon'}{2} \quad (2.38)$$

Como:

$$|A - B| \leq |A| + |B| < \frac{\varepsilon'}{2} + \frac{\varepsilon'}{2} = \varepsilon' \quad (2.39)$$

Substituindo (2.37) e (2.38) em (2.39), temos:

$$\left| \sum_{i=1}^n (W_i \sigma(P_i x + \alpha_i) + \beta_i) - \sum_{i=1}^n (W_i r_n(P_i x + \alpha_i) + \beta_i) - \sum_{i=1}^{2n+1} g_i \left(\sum_{j=1}^n \psi_{ji}(x_j) \right) + \sum_{i=1}^{2n+1} p_{ni} \left(\sum_{j=1}^n q_{nj}(x_j) \right) \right| =$$

$$\left[\sum_{i=1}^n (W_i \sigma(P_i x + \alpha_i) + \beta_i) - \sum_{i=1}^{2n+1} g_i \left(\sum_{j=1}^n \psi_{ji}(x_j) \right) \right] - \left[\sum_{i=1}^n (W_i r_n(P_i x + \alpha_i) + \beta_i) - \sum_{i=1}^{2n+1} p_{ni} \left(\sum_{j=1}^n q_{nj}(x_j) \right) \right] \quad (2.40)$$

Mas,

$$\left| \sum_{i=1}^n (W_i r_n (P_i x + \alpha_i) + \beta_i) - \sum_{i=1}^{2n+1} p_{n_i} \left(\sum_{j=1}^n q_{n_j} (x_j) \right) \right| < \varepsilon \quad (2.41)$$

Pois, p_n , q_n e r_n são seqüências uniformemente convergentes. Logo:

$$\left| \sum_{i=1}^n (W_i \sigma (P_i x + \alpha_i) + \beta_i) - \sum_{i=1}^{2n+1} g_i \left(\sum_{j=1}^n \psi_{j_i} (x_j) \right) \right| - \quad (2.42)$$

$$\left| \sum_{i=1}^n (W_i r_n (P_i x + \alpha_i) + \beta_i) - \sum_{i=1}^{2n+1} p_{n_i} \left(\sum_{j=1}^n q_{n_j} (x_j) \right) \right| \leq$$

$$\leq \left[\left| \sum_{i=1}^n (W_i \sigma (P_i x + \alpha_i) + \beta_i) - \sum_{i=1}^{2n+1} g_i \left(\sum_{j=1}^n \psi_{j_i} (x_j) \right) \right| - \left| \sum_{i=1}^n (W_i r_n (P_i x + \alpha_i) + \beta_i) - \sum_{i=1}^{2n+1} p_{n_i} \left(\sum_{j=1}^n q_{n_j} (x_j) \right) \right| \right] < \varepsilon \quad (2.43)$$

De (2.41), (2.42) e (2.43), vem:

$$\left| \sum_{i=1}^n (W_i \sigma(P_i x + \alpha_i) + \beta_i) - \sum_{i=1}^{2n+1} g_i \left(\sum_{j=1}^n \psi_{ji}(x_j) \right) \right| - \varepsilon'' < \varepsilon' + \varepsilon'' \quad (2.44)$$

$$\left| \sum_{i=1}^{2n+1} g_i \left(\sum_{j=1}^n \psi_{ji}(x_j) \right) - \sum_{i=1}^n (W_i \sigma(P_i x + \alpha_i) + \beta_i) \right| < \varepsilon' + 2\varepsilon'' = \varepsilon \quad (2.45)$$

Donde conclui-se que $y(x)$ converge uniformemente para f .

□

Observação:

O teorema anterior mostra que redes neurais com apenas uma camada intermediária são capazes de aproximar funções reais de um espaço de dimensão finita para o conjunto dos números reais.

O corolário seguinte estende o teorema acima para uma função cuja imagem seja $Y \subset \mathbb{R}^n$.

Corolário 2.3.1

Seja $f: \mathbb{I}^n \rightarrow \mathbb{R}$, e $y_k(x) = \sum_{i=1}^n W_i \sigma(P_i x + \alpha_i) + \beta_i$ uma rede neural, onde n é o número de elementos na camada intermediária, $k = 1, 2, \dots, m$ e m o número de equações de saída. Então $y_k(x)$ converge uniformemente para f .

Prova:

Basta aplicar o teorema anterior para cada $y_j(x)$, onde $j = 1, 2, \dots, m$.

■

2.4. CONCLUSÃO

Os resultados apresentados na seção 2.3, mais precisamente com o teorema (2.3.3) juntamente com o corolário (2.3.1), mostram que redes neurais artificiais são capazes de aproximar arbitrariamente uma função contínua de um espaço de dimensão finita para outro. Entretanto, para se obter a precisão desejada, deve-se levar em consideração o número de elementos necessários na camada intermediária.

CAPÍTULO 3

DIMENSIONAMENTO DE REDES NEURAIS ARTIFICIAIS

3.1 INTRODUÇÃO

Neste capítulo, analisa-se o problema de encontrar o número mínimo de parâmetros de ajuste necessários para que as redes neurais artificiais adquiram as condições de treinamento. São analisados os tipos de redes neurais de várias camadas que utilizam o algoritmo backpropagation, proposto por Rumelhart [1]. Tal rede possui a propriedade de aprendizado através do ajuste de um conjunto de parâmetros chamados de pesos, utilizando um processo de treinamento supervisionado. As redes são treinadas por apresentação de um conjunto de dados, que são chamados de amostras de treinamento $A_{tm}(X, Y)$, onde X, Y representa o conjunto de padrões de entrada e saída, respectivamente. Dessa maneira, as redes vão ajustando seus pesos e seus valores de limiar da ativação para acomodar os

pares (x,y) , onde $x \in X$ e $y \in Y$, representando dessa maneira o mapeamento implícito.

Tais tipos de redes usam como função de ativação a função sigmóide (definição 2.2.3). A informação flui da entrada para a saída, enquanto que o erro da saída atual é comparado com o desejado, propagando-se retroativamente. Dessa maneira, gera-se o espaço dos pesos entre as conexões, que contém um conjunto W onde estão armazenadas as características do mapeamento. Utiliza-se o critério de minimizar o erro quadrático médio e o método do gradiente descendente. Dessa maneira, garante-se que, após um certo período de treinamento, a rede neural seja capaz de descobrir o mapeamento do espaço de entrada para o espaço de saída

Este capítulo é composto de quatro (4) seções. Na seção 2, define-se e representa-se a arquitetura de uma rede neural através da figura 3.1. A a seção 3 caracteriza as condições de treinabilidade e, finalmente na seção 4, apresentam-se as conclusões.

3.2 ARQUITETURA DE REDES NEURAIIS

Definição 3.2.1 (arquitetura de redes neurais) [15]

A arquitetura de uma Rede Neural Artificial (RNA) é composta de uma quintupla

$$ARQ = (x, y, u, v, w) \quad (3.1)$$

onde $x \in X \subseteq \mathbb{R}^{n_i}$, representa a entrada da rede; $y \in Y \subseteq \mathbb{R}^{n_o}$, representa a saída, $u \in U \subseteq \mathbb{R}^n$, $v \in V \subseteq \mathbb{R}^n$, representam a entrada e saída dos elementos da camada intermediária, respectivamente; $w \in W \subseteq \mathbb{R}^q$, representa os parâmetros de ajuste, que podem ser decompostos em quatro submatrizes $w = [P, Q, \alpha, \beta]$, correspondendo aos pesos entre os neurônios da camada de entrada para a camada intermediária e desta para a camada de saída, e os valores de limiar, respectivamente; X, Y, U, V e W são o conjunto de todas as entradas, saídas e parâmetros de ajuste da RNA.

Observação:

A arquitetura de uma rede neural é única definidos, os cinco vetores. Especificamente, x, y determinam as amostras de treinamento da RNA; u e v determinam a entrada e saída dos neurônios respectivamente e w determina todos os parâmetros de ajuste que representam o comportamento funcional da rede.

A figura 3.1 abaixo ilustra a arquitetura de uma RNA.

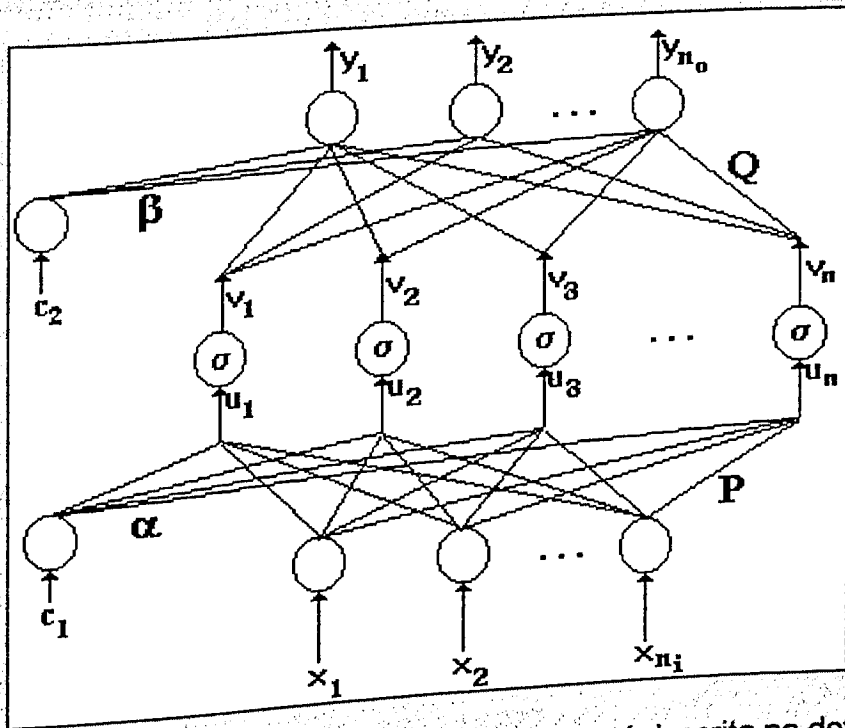


Figura 3.1: Arquitetura de uma rede neural conforme está descrita na definição 3.2.1

A figura 3.1 ilustra a arquitetura de uma rede neural artificial com 3 camadas, sendo uma de entrada, uma intermediária e uma de saída, e P, Q, α , β , formando os parâmetros de ajuste e sendo c_1 e c_2 constantes.

O processo de aprendizado da rede pode ser formalizado da seguinte maneira:

Seja y a(s) saída(s) de uma RNA e considere um número limitado de amostras. O erro entre a saída atual da rede z^p e a saída correta do sistema y^p será dado por $e(z^p, y^p)$ e por uma função de avaliação do desempenho do erro $E(e)$, para os parâmetros de ajuste w , podendo ser calculada segundo uma métrica pré-definida, onde $p=1, 2, 3, \dots, m$, e m é o número de amostras. Otimizando E em relação a w , um vetor w^* pode ser obtido, onde w^* são os parâmetros de ajuste de tal maneira que torne $e(z^p, y^p)$ mínimo.

3.3 TREINABILIDADE

Definição 3.3.1 [15]

Dados uma RNA e um conjunto de amostras de treinamento $A_{tm}=(X, Y)$, diz-se que a RNA é potencialmente treinável se o conjunto de parâmetros de ajuste definidos w^* puder ser obtido, isto é, RNA é potencialmente treinável se $\forall (x, y) \in A_{tm}$, $\forall \varepsilon > 0$, $\exists w^* \in W$, e $p_0 > 0$, tal que $\forall p > p_0$ (p : representa a p -ésima apresentação de um elemento de A_{tm} à rede), tem-se:

$$\|w(p) - w^*\| \leq \varepsilon$$

Definição 3.3.2 [15]

Uma rede neural será treinável se for potencialmente treinável e $E(w) \leq \delta$, onde δ é a tolerância.

Definição 3.3.3 [15]

Uma rede neural será globalmente treinável se ela for treinável para todo $w(0)$ (parâmetro inicial).

Definição 3.3.4 [15]

Uma RNA será invariavelmente treinável se for treinável para todo A_{tm} .

Definição 3.3.5 [15]

Uma RNA será absolutamente treinável com respeito a uma função de avaliação $E[w]$ se for treinável e:

$$E(w^*) = \inf_{w \in W} E[w]$$

onde \inf representa o ínfimo (menor dos valores) do conjunto de valores $E[w]$.

Observação:

A treinabilidade potencial implica na convergência uniforme (definição 2.2.2) da rede que é um pré-requisito para uma aproximação significativa. A treinabilidade absoluta tem como consequência uma aproximação exata. Entretanto a treinabilidade invariante é uma propriedade ideal para as RNAs, que é muito forte para ser satisfeita.

O problema central desse capítulo é estabelecer quais as condições que uma RNA deve apresentar para que a mesma seja treinável em uma das condições acima definidas.

Teorema 3.3.1 [15]

Dado uma RNA e uma função de avaliação $E[w(p)]$, se existir um $M \in \mathbb{R}$ tal que $E[w(p)] > M$ para todo $w \in W$, e $\frac{\partial E[w(p)]}{\partial p} \leq 0$, então a RNA é potencialmente treinável.

Prova:

Pela hipótese $E[w(p)] > M$, e $\frac{\partial E[w(p)]}{\partial p} \leq 0$, nos diz que $E[w(p)]$ é limitado inferiormente por M , e não crescente.

Logo deve existir um ponto w^* tal que, de acordo com a definição (3.3.1):

$$\|w(p) - w^*\| \leq \varepsilon \quad (3.1)$$

Observação:

A figura abaixo mostra a forma aproximada de uma possível função $E[w(p)]$

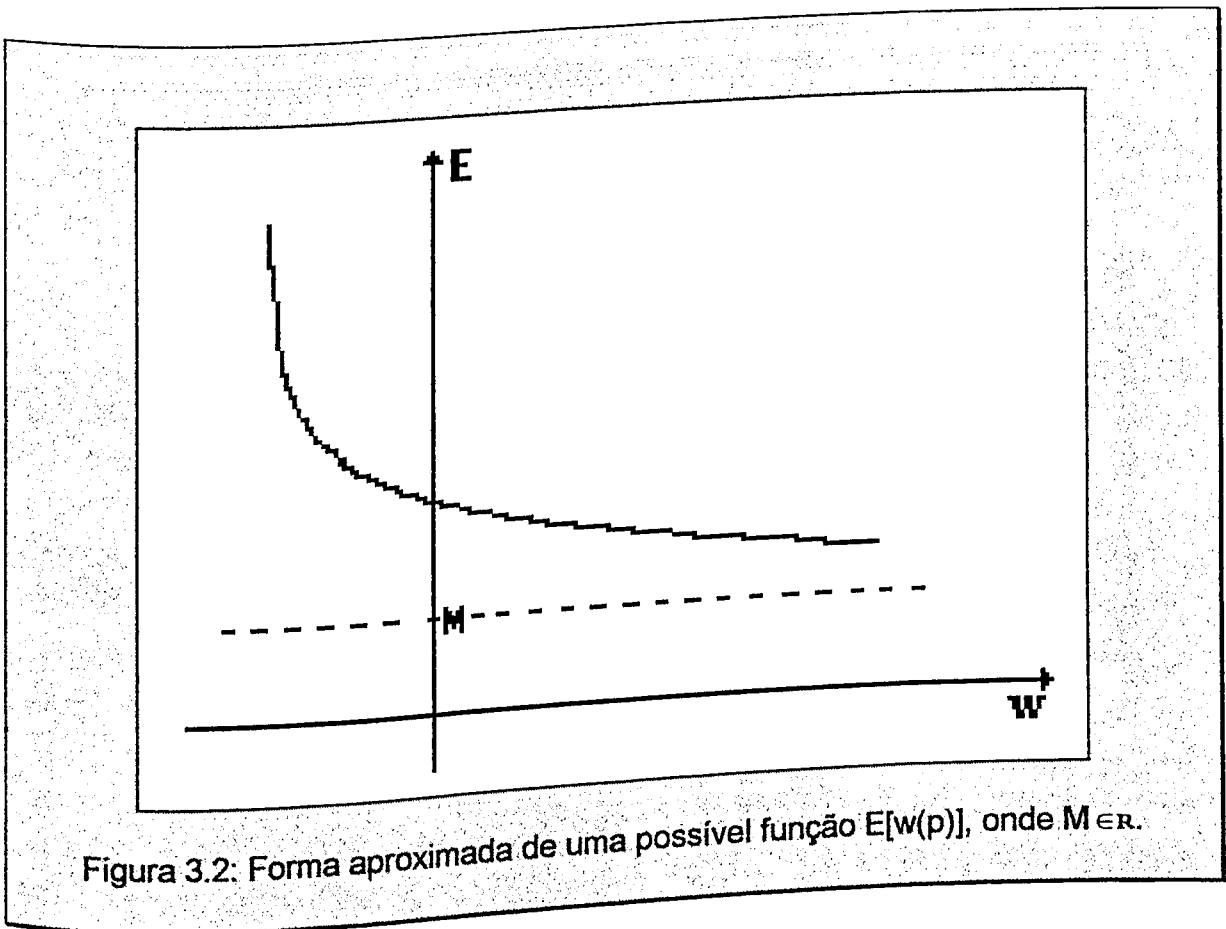


Figura 3.2: Forma aproximada de uma possível função $E[w(p)]$, onde $M \in \mathbb{R}$.

Teorema 3.3.2 [15]

Dado uma RNA e uma função de avaliação $E[w(p)]$, se existe um $M \in \mathbb{R}$ tal que $E[w(p)] > M$ para todo $w \in W$, $\frac{\partial E[w(p)]}{\partial p} \leq 0$, e ainda w^* , com $\frac{\partial E(w^*)}{\partial p} = 0$, então, a RNA é globalmente e absolutamente treinável., se e somente se, $E(w^*) = \inf_{w \in W} E[w(p)]$,

$$\forall w^* \in W, \text{ e } \frac{\partial E[w(p)]}{\partial p} < 0, \text{ para } w \neq w^*,$$

Prova:

Pelo teorema 2.3 e teorema 3.3.1 vem que a RNA é globalmente treinável.

Ainda pelo teorema 3.3.1 deve existir w^* , satisfazendo a equação (3.1). Se $E(w^*) = \inf_{w \in W} E[w(p)]$, $\forall w^* \in W$, e $\frac{\partial E[w(p)]}{\partial p} < 0$, para $w \neq w^*$, pela definição 3.3.5,

vem que a RNA é absolutamente treinável.

O que mostra que a RNA é globalmente e absolutamente treinável.

Corolário 3.3.1 [15]

Dada uma função de avaliação limitada inferiormente, a RNA com a regra de aprendizagem baseada no gradiente da função é potencialmente treinável.

Prova:

$$\frac{\partial w}{\partial p} = -\eta(p) \nabla_w^T E[w(p)] \quad (3.2)$$

onde $\eta(p) > 0$, e $\nabla_w^T E[w(p)]$ é o vetor gradiente de $E[w(p)]$

De acordo com a equação (3.2), vem:

$$\nabla_w^T E[w(p)] = -\frac{1}{\eta(p)} \frac{\partial w}{\partial p} \quad (3.3)$$

Multiplicando ambos os membros da equação (3.3) por: $\left[\frac{\partial w}{\partial p} \right]^T$

$$\left(\frac{\partial w}{\partial p} \right)^T \nabla_w^T E[w(p)] = -\frac{1}{\eta(p)} \left[\left(\frac{\partial w}{\partial p} \right)^T \frac{\partial w}{\partial p} \right] \quad (3.4)$$

Da equação (3.4), vem:

$$\nabla_w E[w(p)] \left(\frac{\partial w}{\partial p} \right) = -\frac{1}{\eta(p)} \left[\frac{\partial w}{\partial p} \left(\frac{\partial w}{\partial p} \right)^T \right] \quad (3.5)$$

Donde, se obtém:

$$-\frac{1}{\eta(p)} \left[\frac{\partial w}{\partial p} \left(\frac{\partial w}{\partial p} \right)^T \right] \leq 0 \quad (3.6)$$

Por outro lado, tem-se:

$$\nabla_w E[w(p)] \left(\frac{\partial w}{\partial p} \right) = \frac{\partial E[w(p)]}{\partial p} \quad (3.7)$$

De (3.5), (3.6) e (3.7), vem:

$$\frac{\partial E[w(p)]}{\partial p} \leq 0 \quad (3.8)$$

Como $E[w(p)]$ é limitada inferiormente e vale (3.8), do teorema (3.3.1), conclui-se que a RNA é potencialmente treinável.

Teorema 3.3.3 [15]

Seja uma função de avaliação dada por:

$$E[w] = \frac{1}{n_i k} \sum_{j=1}^{n_i} \mu_j \sum_{i=1}^{n_o} |e_i^j(x^j, w)|^k \quad (3.9)$$

onde $e_i^j(x^j, w) = z_i^j - y_i(x_j, w)$ é o erro da saída da rede e $1 < k < +\infty$. Uma RNA com uma função de ativação diferenciável e com a regra de aprendizagem baseada no gradiente da função de avaliação será globalmente e absolutamente treinável e assumirá um mínimo global em $E[w(p)]=0$, se e somente se, o posto da matriz $F(x, w)$ for $n_i \cdot n_o$, isto é,

$$\text{posto}(\Phi(x, w)) = n_i \cdot n_o \quad (3.10)$$

onde

$$\Phi(x, w) = \begin{pmatrix} \frac{\partial y_1(x^1, w)}{\partial w_1} & \frac{\partial y_1(x^1, w)}{\partial w_2} & \dots & \frac{\partial y_1(x^1, w)}{\partial w_q} \\ \frac{\partial y_1(x^2, w)}{\partial w_1} & \frac{\partial y_1(x^2, w)}{\partial w_2} & \dots & \frac{\partial y_1(x^2, w)}{\partial w_q} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial y_1(x^{n_i}, w)}{\partial w_1} & \frac{\partial y_1(x^{n_i}, w)}{\partial w_2} & \dots & \frac{\partial y_1(x^{n_i}, w)}{\partial w_q} \\ \frac{\partial y_2(x^1, w)}{\partial w_1} & \frac{\partial y_2(x^1, w)}{\partial w_2} & \dots & \frac{\partial y_2(x^1, w)}{\partial w_q} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial y_{n_o}(x^{n_i}, w)}{\partial w_1} & \frac{\partial y_{n_o}(x^{n_i}, w)}{\partial w_2} & \dots & \frac{\partial y_{n_o}(x^{n_i}, w)}{\partial w_q} \end{pmatrix} \quad (3.11)$$

n_i é a dimensão da camada de entrada da rede, n_o é a dimensão da camada de saída, x^j é o j -ésimo padrão de treinamento e q representa a dimensão do espaço dos parâmetros de ajuste.

Corolário 3.3.2 [15]

Se uma RNA é, globalmente e absolutamente treinável, onde q representa o número de parâmetros de ajuste w , n_i o número de padrões de treinamento e n_o o número de padrões de saída então $q \geq n_i \cdot n_o$.

Prova:

Como A RNA é absolutamente e globalmente treinável, pelo teorema 3.3.3 equações (3.10) e (3.11), segue-se que $q \geq n_i \cdot n_o$.

■

3.4 CONCLUSÃO

Neste capítulo foram analisadas as condições de treinabilidade para as RNAs estabelecidas em [15]. Isto foi mostrado através do corolário 3.3.2, onde se define um número mínimo de parâmetros de ajuste que uma RNA deve ter para que a mesma possa ser treinável em uma das condições definidas. Como se pode observar, ainda não foram estabelecidas as condições para que se calcule o

número de elementos na camada intermediária de uma RNA dado um conjunto de amostras de treinamento A_{trn} . Esta é a principal contribuição deste trabalho e será objeto do capítulo posterior.

CAPÍTULO 4

O USO DO TEOREMA DA FUNÇÃO IMPLÍCITA NO DIMENSIONAMENTO DE REDES NEURAIS MULTI-CAMADAS

4.1 INTRODUÇÃO

Neste capítulo, são estabelecidas condições para o dimensionamento de redes neurais artificiais de várias camadas, usadas para aproximação de mapeamentos não-lineares. As redes neurais são tratadas como sistemas de equações algébricas, não-lineares, tendo como incógnitas os pesos das conexões. Usa-se então o teorema da função implícita para demonstrar as condições estabelecidas. Estas condições foram propostas em [18] e são, formalmente, demonstradas neste capítulo.

Redes neurais de várias camadas [1,2] têm sido usadas em aplicações tais como reconhecimento de padrões e identificação de mapeamentos não-lineares. Teoricamente, a aplicação de redes neurais para aproximação funcional pode ser analisada usando-se o teorema de Wierstrass [3]. Se a função de ativação de cada elemento da rede neural for uma função contínua de valor real, então, será possível fazer a aproximação polinomial e conseqüentemente a relação funcional de entrada-saída da rede poderá ser aproximada por um polinômio.

A precisão da aproximação funcional está diretamente relacionada com a estrutura da rede neural [4,5,6]. As seguintes causas de imprecisão devem ser consideradas no emprego de redes neurais para aproximação de mapeamentos não-lineares: (a) O número de padrões usados para treinar a rede pode não ser suficiente para captar a estrutura real do mapeamento; (b) A rede neural poderá estar subdimensionada, não possuindo número suficiente de elementos na camada intermediária e, portanto, a tolerância desejada não poderá ser atingida. A primeira depende do conhecimento a respeito do mapeamento a ser ensinado. A segunda fonte de imprecisão pode ser evitada pela escolha correta da arquitetura da rede neural. Para isso, são estabelecidos limites para o número de elementos das camadas intermediárias. Isto é enfatizado neste capítulo, iniciando com algumas definições matemáticas e afirmações, além de se formalizar a representação da rede neural. Em seguida, o teorema proposto em [18] é provado, para estabelecer o número de elementos necessários da camada intermediária para produzir a aproximação funcional desejada.

4.2 DEFINIÇÕES

As seguintes definições são necessárias para o estabelecimento das condições de estudo das redes neurais.

Definição 4.2.1

Um elemento da rede neural é um dispositivo de processamento cuja saída y é o resultado da soma ponderada de uma entrada x e uma referência unitária, calculada através da função de ativação σ , isto é:

$$y(x) = \sigma \left(\sum w x + \theta \right) \quad (4.1)$$

onde $w \in W \subset \mathbb{R}^q$ é um vetor contendo o valor das conexões e θ é o valor de limiar.

Definição 4.2.2

Uma rede neural de alimentação direta é uma estrutura com, ou sem, camadas intermediárias onde as saídas de cada um dos elementos de uma camada estão conectados a todos os elementos das camadas subseqüentes incluindo a

saída. Uma classe particular de redes neurais é aquela que possui conexões apenas entre as camadas adjacentes.

Observação:

Embora diversas funções possam ser usadas como a função de ativação, a função sigmóide (definição 2.2.2) é a mais comumente usada pelo fato de ser monótona e crescente. Estas características são importantes para satisfazer as condições de treinabilidade da rede neural (Veja o Teorema 3.3.3).

4.3 EQUAÇÕES DA REDE NEURAL

Considere inicialmente uma rede neural com somente uma camada intermediária com n elementos. Seja y a saída da rede neural treinada com os padrões de entrada-saída $\{[kt, y(k)], k=1, \dots, m\}$. Suponha que f seja uma função contínua, amostrada uniformemente com período t , isto é, $\{y(k)=f(kt), k=1, \dots, m\}$. Se n for o número de elementos na camada intermediária, a saída da rede será:

$$y(x) = \sigma \left(\sum_{k=1}^n Q_k \sigma (P_k x + \alpha_k) + \beta \right) \quad (4.2)$$

onde s é a função de ativação, P_k representa os pesos das conexões da entrada para a camada intermediária e Q_k representa os pesos das conexões da camada intermediária para a saída; α_k e β representam os valores de limiares da camada intermediária e da camada de saída, respectivamente. (vide figura 3.1).

A saída da rede neural para um ponto dentro do intervalo amostragem (interpolação) pode ser calculada por:

$$y(x + \Delta x) = \sigma \left(\sum_{k=1}^n Q_k \sigma (P_k (x + \Delta x) + \alpha_k) + \beta \right) \quad (4.3)$$

onde: σ é a função sigmóide e Δx é a distância do ponto em teste ao ponto amostrado mais próximo. O somatório na equação (4.3) representa a entrada líquida do elemento de saída da rede neural, isto é:

$$z(x) = \sum_{k=1}^n Q_k \frac{1}{1 + e^{-(P_k x + \alpha_k)}} \quad (4.4)$$

A derivada da equação (4.4) é dada por:

$$\frac{\partial z(x)}{\partial x} = z(x) - [1 - z(x)] \quad (4.5)$$

A equação 4.5 acima citada decorre da proposição 2.2,1 e verifica-se que o valor máximo da derivada dado em (4.5) depende dos valores dos pesos. Assim sendo, a amplitude máxima das oscilações entre dois pontos amostrados estará limitada por:

$$y(x + \Delta x)_{MAX} = \sigma \left[\left(\frac{\partial z(x)}{\partial x} \right) \Delta x + \beta \right] \quad (4.6)$$

A equação (4.6) mostra que, dentro do intervalo de amostragem, a amplitude da aproximação pode ser inaceitável dependendo dos pesos das conexões. Verifica-se (por meio das simulações realizadas) também que se o número de elementos da camada intermediária for aumentado, haverá, como conseqüência, uma maior tendência para oscilações indesejáveis. Apesar das amplitudes de tais oscilações serem imprevisíveis, o índice de Nyquist [19] produz uma diretriz bastante útil para se determinar o número de padrões de treinamento. Depois do processo de aprendizagem, deve-se testar a rede neural em pelo menos um ponto dentro e cada intervalo de amostragem. Se o erro de saída não corresponder à tolerância especificada, então o número de amostras deverá ser acrescido. Observa-se contudo que, quando se aumenta o número de amostras, há necessidade de se aumentar o número de elementos da camada intermediária (veja teorema a seguir).

Teorema 4.3.1 [18]

Dadas m padrões de treinamentos distintos, o número suficiente de parâmetros de ajuste em uma rede neural com várias entradas, uma saída e uma camada intermediária de n elementos, capaz de representar exatamente a relação de entrada-saída, é igual a m , considerando que a função de ativação seja monótona e de classe C' em \mathbb{R}^n .

Prova:

Considere uma rede neural com n_i entradas uma saída e uma única camada intermediária com n elementos. O conjunto de equações de saída, para m amostras é:

$$y(1) - \sigma \left(\begin{array}{l} Q_1 \sigma \left(P_{1,1} \cdot x_1(1) + \dots + P_{1,n_i} \cdot x_{n_i}(1) + \alpha_1 \right) + \dots \\ + Q_n \sigma \left(P_{n,1} \cdot x_1(1) + \dots + P_{n,n_i} \cdot x_{n_i}(1) + \alpha_1 \right) + \beta \end{array} \right) = 0$$

(4.7)

$$y(m) - \sigma \left(\begin{array}{l} Q_1 \sigma \left(P_{1,1} \cdot x_1(m) + \dots + P_{1,n_i} \cdot x_{n_i}(m) + \alpha_1 \right) + \dots \\ + Q_n \sigma \left(P_{n,1} \cdot x_1(m) + \dots + P_{n,n_i} \cdot x_{n_i}(m) + \alpha_1 \right) + \beta \end{array} \right) = 0$$

onde $P_{i,j}$ é o peso da conexão entre a entrada j e o elemento i da camada intermediário; Q_k é o peso da conexão entre o k -ésimo elemento da camada intermediária e a saída; α_k é o valor de limiar do k -ésimo elemento, β é o valor de limiar da saída.

Para simplificar a notação, seja o vetor $w \in W \subset \mathbb{R}^q$ representando todos os parâmetros de ajuste ou pesos de uma rede neural, isto é,

$$w = (P_{1,1}, P_{1,2}, \dots, P_{1,n_1}, P_{2,1}, \dots, P_{2,n_2}, \dots, P_{n,1}, \dots, P_{n,n_n}, Q_1, \dots, Q_n, \alpha_1, \dots, \alpha_n, \beta) \quad (4.8)$$

Seja, o conjunto de equações (4.7) representado por:

$$\begin{cases} \phi_1(w_1, w_2, \dots, w_q) = 0 \\ \dots \\ \phi_m(w_1, w_2, \dots, w_q) = 0 \end{cases} \quad (4.9)$$

onde ϕ_1, \dots, ϕ_m são de classe C^1 , pois são composições de funções sigmóides

(proposição 2.2.1), e $q = r(n+2) + 1$.

Seja a seguinte transformação $T: \mathbb{R}^q \rightarrow \mathbb{R}^q$ dada por:

$$T = \begin{cases} t_1 = w_1 \\ t_2 = w_2 \\ t_3 = w_3 \\ \cdot \\ \cdot \\ \cdot \\ t_{q-m+1} = \Phi_1(w_1, w_2, \dots, w_q) \\ t_{q-m+2} = \Phi_2(w_1, w_2, \dots, w_q) \\ t_{q-m+3} = \Phi_3(w_1, w_2, \dots, w_q) \\ \cdot \\ \cdot \\ \cdot \\ t_q = \Phi_m(w_1, w_2, \dots, w_q) \end{cases} \quad (4.10)$$

Montando o Jacobiano da transformação (4.10), tem-se:

$$\text{Jacob}(T) = \begin{pmatrix}
 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 0 & \dots & 1 & 0 & 0 & 0 & \dots & 0 \\
 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\
 \frac{\partial \phi_1}{\partial w_1} & \frac{\partial \phi_1}{\partial w_2} & \frac{\partial \phi_1}{\partial w_3} & \dots & \frac{\partial \phi_1}{\partial w_{q-m+1}} & \frac{\partial \phi_1}{\partial w_{q-m+2}} & \dots & \frac{\partial \phi_1}{\partial w_q} \\
 \frac{\partial \phi_2}{\partial w_1} & \frac{\partial \phi_2}{\partial w_2} & \frac{\partial \phi_2}{\partial w_3} & \dots & \frac{\partial \phi_2}{\partial w_{q-m+1}} & \frac{\partial \phi_2}{\partial w_{q-m+2}} & \dots & \frac{\partial \phi_2}{\partial w_q} \\
 \frac{\partial \phi_m}{\partial w_1} & \frac{\partial \phi_m}{\partial w_2} & \frac{\partial \phi_m}{\partial w_3} & \dots & \frac{\partial \phi_m}{\partial w_{q-m+1}} & \frac{\partial \phi_m}{\partial w_{q-m+2}} & \dots & \frac{\partial \phi_m}{\partial w_q}
 \end{pmatrix} \quad (4.11)$$

De (4.11) [11], tem-se:

$$\text{Det}[\text{jacob}(T)] = \begin{vmatrix}
 1 & 0 & \dots & 0 \\
 0 & 1 & \dots & 0 \\
 \dots & \dots & \dots & \dots \\
 0 & 0 & \dots & 1
 \end{vmatrix} \cdot \begin{vmatrix}
 \frac{\partial \phi_1}{\partial w_{q-m+1}} & \frac{\partial \phi_1}{\partial w_{q-m+2}} & \dots & \frac{\partial \phi_1}{\partial w_q} \\
 \frac{\partial \phi_2}{\partial w_{q-m+1}} & \frac{\partial \phi_2}{\partial w_{q-m+2}} & \dots & \frac{\partial \phi_2}{\partial w_q} \\
 \dots & \dots & \dots & \dots \\
 \frac{\partial \phi_m}{\partial w_{q-m+1}} & \frac{\partial \phi_m}{\partial w_{q-m+2}} & \dots & \frac{\partial \phi_m}{\partial w_q}
 \end{vmatrix} \quad (4.12)$$

Pelo Lema 4.3.1 (abaixo), e considerando que os padrões são todos distintos, tem-se:

$$\begin{vmatrix}
 \frac{\partial \phi_1}{\partial w_{q-m+1}} & \frac{\partial \phi_1}{\partial w_{q-m+2}} & \cdots & \frac{\partial \phi_1}{\partial w_q} \\
 \frac{\partial \phi_2}{\partial w_{q-m+1}} & \frac{\partial \phi_2}{\partial w_{q-m+2}} & \cdots & \frac{\partial \phi_2}{\partial w_q} \\
 \frac{\partial \phi_m}{\partial w_{q-m+1}} & \frac{\partial \phi_m}{\partial w_{q-m+2}} & \cdots & \frac{\partial \phi_m}{\partial w_q}
 \end{vmatrix} \neq 0 \quad (4.13)$$

Por (4.12) e (4.13) ve-se que:

$$\text{Det}[\text{Jacob}(\Gamma)] \neq 0 \quad (4.14)$$

Pelo teorema da função implícita [9], dado o conjunto de equações (4.9) com q variáveis, e um ponto:

$$w^0 = (\bar{w}_1, \bar{w}_2, \dots, \bar{w}_q) \quad (4.15)$$

que satisfaça o sistema de equações (4.9), pode-se, teoricamente encontrar funções

$\psi_{q-m+1}, \dots, \psi_q$, tais que:

$$\begin{cases}
 \phi_1(w_1, w_2, \dots, w_{q-m}, \psi_{q-m+1}, \dots, \psi_q) = 0 \\
 \dots \\
 \phi_m(w_1, w_2, \dots, w_{q-m}, \psi_{q-m+1}, \dots, \psi_q) = 0
 \end{cases} \quad (4.16)$$

onde $\psi_{q-m+1}, \psi_{q-m+2}, \dots, \psi_q$ dependem de w_1, w_2, \dots, w_{q-m} , sejam soluções do sistema dado, para um conjunto de variáveis, w_{q-m+1}, \dots, w_q , em termos das variáveis restantes, numa vizinhança de w° . ■

Para as redes neurais de múltiplas saídas, o conjunto de equações (4.9) deve ser multiplicado pelo número de saídas. O corolário seguinte formaliza a condição de existência para o caso de várias saídas.

Lema 4.3.1

Sejam $\phi_1, \phi_2, \dots, \phi_m : W \subset \mathbb{R}^q \rightarrow \mathbb{R}$, funções definidas por composições de funções sigmóides, e ainda $\phi_1 \neq \phi_2 \neq \dots \neq \phi_m$ então o determinante do jacobiano das funções ϕ_i , $i = 1, 2, \dots, m$ será não nulo.

Prova:

Montando o conjunto de equações de ϕ , tem-se:

$$S = \begin{cases} \phi_1(w_1, w_2, \dots, w_q) \\ \phi_2(w_1, w_2, \dots, w_q) \\ \vdots \\ \phi_m(w_1, w_2, \dots, w_q) \end{cases} \quad (4.17)$$

Montando o jacobiano de (4.17), tem-se:

$$Jacob[S] = \begin{pmatrix} \frac{\partial \phi_1}{\partial w_1} & \frac{\partial \phi_1}{\partial w_2} & \dots & \frac{\partial \phi_1}{\partial w_q} \\ \frac{\partial \phi_2}{\partial w_1} & \frac{\partial \phi_2}{\partial w_2} & \dots & \frac{\partial \phi_2}{\partial w_q} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \ddot{\phi}_m}{\partial w_1} & \frac{\partial \ddot{\phi}_m}{\partial w_2} & \dots & \frac{\partial \ddot{\phi}_m}{\partial w_q} \end{pmatrix} \quad (4.18)$$

Calculando o determinante de (4.18), segue-se:

(i) $\frac{\partial \phi_1}{\partial w_1} \neq 0$, por hipótese.

(ii) Suponha que vale (4.19) abaixo:

$$\begin{vmatrix} \frac{\partial \phi_1}{\partial w_1} & \frac{\partial \phi_1}{\partial w_2} & \dots & \frac{\partial \phi_1}{\partial w_q} \\ \frac{\partial \phi_2}{\partial w_1} & \frac{\partial \phi_2}{\partial w_2} & \dots & \frac{\partial \phi_2}{\partial w_q} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \ddot{\phi}_m}{\partial w_1} & \frac{\partial \ddot{\phi}_m}{\partial w_2} & \dots & \frac{\partial \ddot{\phi}_m}{\partial w_q} \end{vmatrix} \neq 0 \quad (4.19)$$

(iii) Temos que provar que vale (4.20):

$$\begin{vmatrix} \frac{\partial \phi_1}{\partial w_1} & \frac{\partial \phi_1}{\partial w_2} & \dots & \frac{\partial \phi_1}{\partial w_q} & \frac{\partial \phi_1}{\partial w_{q+1}} \\ \frac{\partial \phi_2}{\partial w_1} & \frac{\partial \phi_2}{\partial w_2} & \dots & \frac{\partial \phi_2}{\partial w_q} & \frac{\partial \phi_2}{\partial w_{q+1}} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial \phi_m}{\partial w_1} & \frac{\partial \phi_m}{\partial w_2} & \dots & \frac{\partial \phi_m}{\partial w_q} & \frac{\partial \phi_m}{\partial w_{q+1}} \\ \frac{\partial \phi_{m+1}}{\partial w_1} & \frac{\partial \phi_{m+1}}{\partial w_2} & \dots & \frac{\partial \phi_{m+1}}{\partial w_q} & \frac{\partial \phi_{m+1}}{\partial w_{q+1}} \end{vmatrix} \neq 0 \quad (4.20)$$

Seja a seguinte matriz A:

$$A = \begin{pmatrix} \frac{\partial \phi_1}{\partial w_1} & \frac{\partial \phi_1}{\partial w_2} & \dots & \frac{\partial \phi_1}{\partial w_q} & \frac{\partial \phi_1}{\partial w_{q+1}} \\ \frac{\partial \phi_2}{\partial w_1} & \frac{\partial \phi_2}{\partial w_2} & \dots & \frac{\partial \phi_2}{\partial w_q} & \frac{\partial \phi_2}{\partial w_{q+1}} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial \phi_m}{\partial w_1} & \frac{\partial \phi_m}{\partial w_2} & \dots & \frac{\partial \phi_m}{\partial w_q} & \frac{\partial \phi_m}{\partial w_{q+1}} \\ \frac{\partial \phi_{m+1}}{\partial w_1} & \frac{\partial \phi_{m+1}}{\partial w_2} & \dots & \frac{\partial \phi_{m+1}}{\partial w_q} & \frac{\partial \phi_{m+1}}{\partial w_{q+1}} \end{pmatrix} \quad (4.21)$$

Pelo teorema de Laplace [20] para determinante de ordem qualquer tem-se que o determinante 4.21, pode ser calculado por:

$$\sum_{i=1}^{m+1} (-1)^{i+q+1} \cdot \frac{\partial \phi_i}{\partial w_{q+1}} \cdot \det[A_{i,q+1}] \quad (4.22)$$

Como $\frac{\partial \phi_i}{\partial w_{q+1}} \neq 0$ por hipótese (pois ϕ_i são composições de funções sigmóides) e por (ii) vemos que (4.22) é não nulo, o que conclui a demonstração do lema. ■

Observação:

Para uma melhor compreensão. Seja uma matriz quadrada de ordem 2, montada conforme 4.18.

$$B = \begin{pmatrix} \frac{\partial \phi_1}{\partial w_1} & \frac{\partial \phi_1}{\partial w_2} \\ \frac{\partial \phi_2}{\partial w_1} & \frac{\partial \phi_2}{\partial w_2} \end{pmatrix} \quad (4.23)$$

Calculando o determinante da matriz B, tem-se:

$$\det[B] = \begin{vmatrix} \frac{\partial \phi_1}{\partial w_1} & \frac{\partial \phi_1}{\partial w_2} \\ \frac{\partial \phi_2}{\partial w_1} & \frac{\partial \phi_2}{\partial w_2} \end{vmatrix} = \frac{\partial \phi_1}{\partial w_1} \cdot \frac{\partial \phi_2}{\partial w_2} - \frac{\partial \phi_1}{\partial w_2} \frac{\partial \phi_2}{\partial w_1} \quad (4.24)$$

Como ϕ_1 e ϕ_2 são composições de funções sigmóides, tem-se:

$$\det[B] \neq 0$$

(4.25)

Desde que $\phi_1 \neq \phi_2$.

■

No teorema 4.3.1 acima a hipótese de que os padrões de treinamento sejam todos distintos é fundamental para que as funções ϕ_i , sejam distintas, assegurando-se assim a independência linear da matriz do jacobiano das funções ϕ .

Corolário 4.3.1

Dado um conjunto com m padrões de treinamento distintos com n_i entradas e n_o saídas, o número suficiente de parâmetros de ajuste em uma rede neural com uma única camada intermediária, para representar exatamente a relação de entrada-saída, para o conjunto de amostras de treinamento apresentado (A_{trn}), é dado pelo produto de n_o por m .

Prova:

Aplicando o teorema 4.3.1 para cada uma das y_j , $j = 1, 2, 3, \dots, n_o$ saídas, vemos que deveríamos ter $m + m + \dots + m$ (n_o vezes) parâmetros de ajuste.

Logo o número suficiente de parâmetros de ajustes será $n_o m$. ■

Corolário 4.3.2

Dado um conjunto de m padrões de treinamento distintos, o número suficiente n de elementos na camada intermediária em uma rede neural de uma única camada intermediária, capaz de representar exatamente a relação de entrada saída é:

$$n = \text{inteiro} \left[\frac{m - n_o}{n_i + n_o + 1} \right] \quad (4.25)$$

onde n_i é a dimensão da entrada e n_o é a dimensão de saída da rede neural.

Prova:

Pelo teorema 4.3.1 para que a rede possa ser absolutamente treinável é necessário que $q=m$, então:

$$m = n(n_i + n_o + 1) + n_o \quad (4.26)$$

Logo de (4.26), segue-se:

$$n = \frac{m - n_o}{n_i + n_o + 1} \quad (4.27)$$

Como n deve ser inteiro toma-se:

$$n = \text{inteiro} \left[\frac{m - n_o}{n_i + n_o + 1} \right]$$

onde inteiro representa o primeiro inteiro superior ou igual a $\frac{m - n_o}{n_i + n_o + 1}$.

Corolário 4.3.3

Dado um conjunto de m padrões distintos, o número suficiente de elementos nas camadas intermediárias em uma rede neural de duas camadas intermediárias,

$$m = n(n_i + n_o + 1) + n_o \quad (4.26)$$

Logo de (4.26), segue-se:

$$n = \frac{m - n_o}{n_i + n_o + 1} \quad (4.27)$$

Como n deve ser inteiro toma-se:

$$n = \text{inteiro} \left[\frac{m - n_o}{n_i + n_o + 1} \right]$$

onde inteiro representa o primeiro inteiro superior ou igual a $\frac{m - n_o}{n_i + n_o + 1}$.

■

Corolário 4.3.3

Dado um conjunto de m padrões distintos, o número suficiente de elementos nas camadas intermediárias em uma rede neural de duas camadas intermediárias,

para representar exatamente a relação de entrada-saída, deve-se satisfazer a seguinte relação:

$$n_o m \leq n_o(n_2 + 1) + n_1(n_i + 1) + n_2(n_1 + 1) \quad (4.28)$$

onde n_i é a dimensão de entrada e n_o a dimensão da saída, n_1 e n_2 são os números de elementos da primeira e segunda camadas intermediárias, respectivamente.

Prova:

Seja uma rede neural com n_i elementos na camada de entrada, n_o elementos na camada de saída, n_1 e n_2 elementos da primeira e segunda camadas intermediárias, respectivamente.

Logo a dimensão da rede será:

$$q = n_o(n_2 + 1) + n_1(n_i + 1) + n_2(n_1 + 1) \quad (4.29)$$

Pelo corolário 4.3.1, deve-se ter:

$$n_o m \leq q \quad (4.30)$$

De 4.29 e 4.30 tem-se:

$$n_o m \leq n_o(n_2 + 1) + n_1(n_i + 1) + n_2(n_1 + 1)$$

■

4.4 O EFEITO DA TOLERÂNCIA DURANTE O TREINAMENTO DA REDE NEURAL

Considere que haja q padrões de treinamento e que a rede neural tenha sido corretamente dimensionada à luz do corolário 4.3.1. Considere também que se admita uma tolerância ϵ para a convergência da rede. Isto equivale, como ilustra a figura 4.1, a admitir que, em torno do valor correto, há infinitos valores aceitáveis de w . Isto é o mesmo que dizer que há infinitos padrões de treinamento. Neste caso, o dimensionamento da rede, com base no teorema 4.3.1 não levará o treinamento a um mínimo global [20]. Isto pode ser observado nas simulações apresentadas no próximo capítulo.

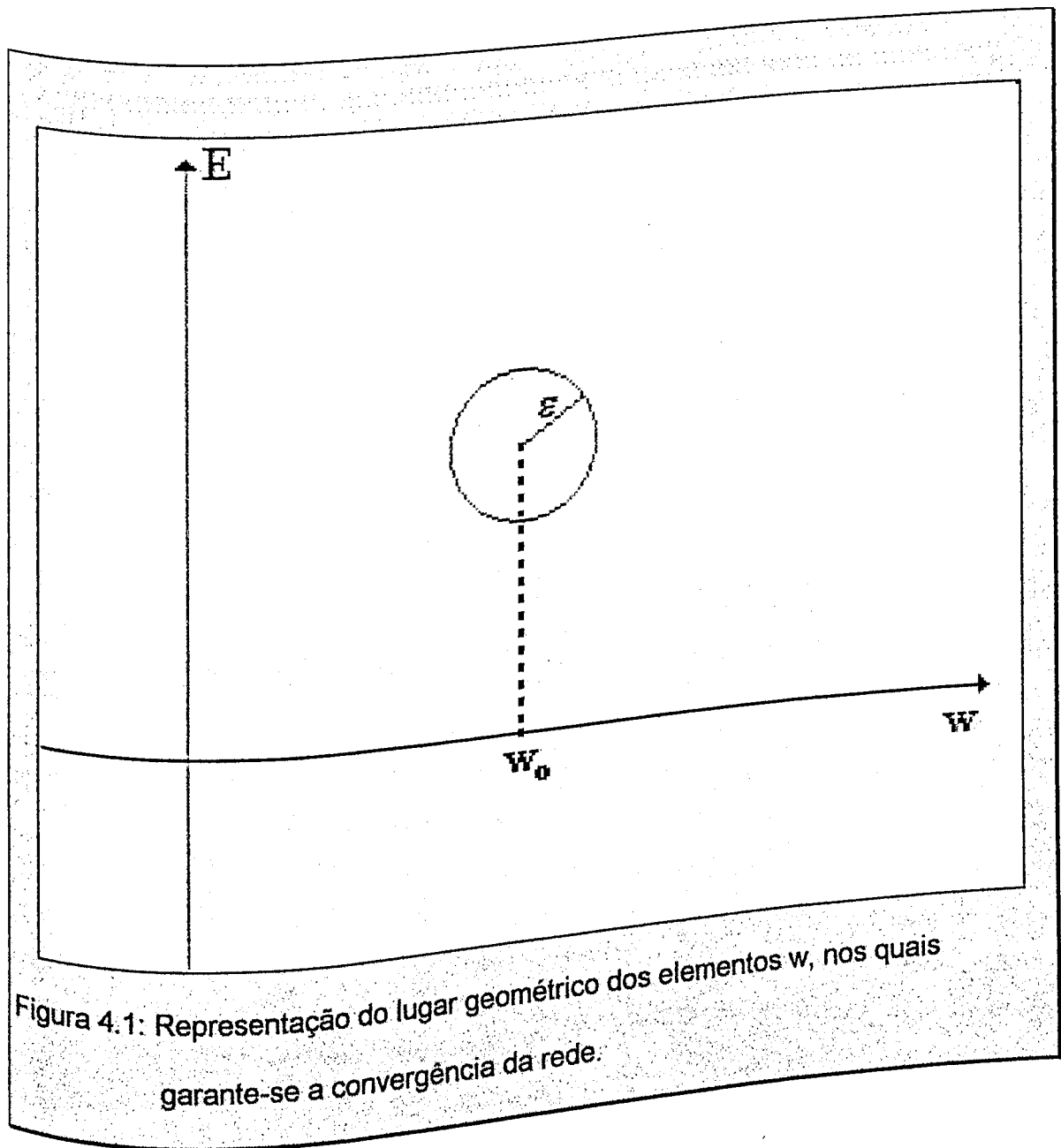


Figura 4.1: Representação do lugar geométrico dos elementos w , nos quais garante-se a convergência da rede.

4.5 CONCLUSÃO

A análise matemática realizada neste capítulo comprova que se o número de padrões de treinamento distintos for igual ao número de parâmetros de ajuste da rede neural, esta será absolutamente treinável. Para o caso $q > m$, haverá infinitas soluções possíveis, em outras palavras pode-se dizer que a rede é potencialmente treinável (definição 3.3.2). Como as funções de ativação são monótonas e de classe C^1 , isto garante que as derivadas parciais da função de ativação nunca se anulam. No caso em que $q < m$ não haverá a convergência da rede (desconsiderando-se a possibilidade de exemplos falsos). Isto está ilustrado mais facilmente nas simulações do próximo capítulo.

Do teorema 4.3.1 e do corolário 4.3.1 é possível concluir que a existência de uma solução não depende do número de camadas mas do número de parâmetros de ajuste, que são os pesos das conexões. Esta conclusão também foi demonstrada por Hornik [7] através do teorema 2.3.3. Conseqüentemente, pode-se definir um número suficiente de elementos na camada intermediária baseado no teorema 4.3.1. Os corolários 2 e 3 especificam o número de elementos para redes de uma e duas camadas, respectivamente, no capítulo posterior foram feitas simulações para se comprovar os resultados teóricos acima, apresentados.

CAPÍTULO 5

SIMULAÇÕES E CONCLUSÃO GERAL

5.1 SIMULAÇÕES

Neste capítulo, apresentam-se alguns exemplos para testar o método de se avaliar o número de amostras e o número de unidades das camadas intermediárias em uma determinada arquitetura de rede neural, para aproximação funcional. Para ilustrar, os procedimentos formulados, treinou-se uma rede neural, como descrito nos exemplos 1 e 2, os valores de erro no treinamento encontram-se no apêndice, nas tabelas A, B e C, onde as tabelas A e B, referem-se ao exemplo 1, casos 1 e 2 respectivamente e a tabela C, refere-se ao exemplo 2.

Exemplo 1 (OU-EXCLUSIVO)

Considere o problema ou-exclusivo (XOR) $f_1(x_1, x_2) = x_1 \oplus x_2$ (lê-se x_1 operado com x_2), onde $x_i \in \{0,1\}$, $i = 1, 2$, $x_1 \otimes x_2 \in \{0,1\}$, a operação acima está definida na tabela 5.1. Duas situações são apresentadas.

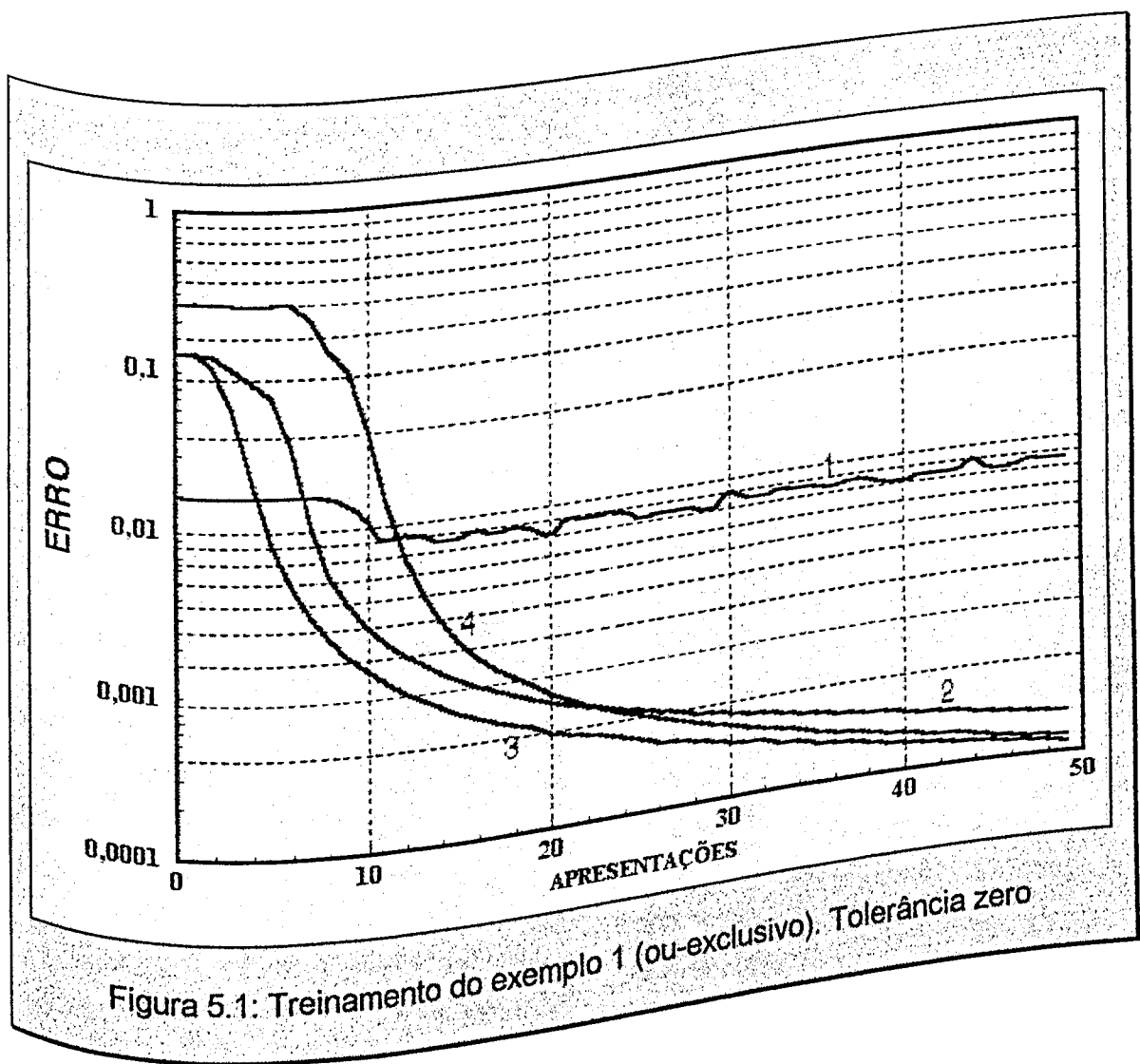
Caso 1

Neste caso, apenas quatro padrões são apresentado à rede. Como se pode observar na figura 5.1, as curvas de 1 a 4 referem-se ao estado onde a rede é treinada com os padrões (0,0), (0,1), (1,0) e (1,1) e com o número de elementos na camada intermediária sendo igual a 1, 2, 10 e 30 respectivamente.

Os resultados estão apresentados na figura 5.1, onde as curvas 1, 2, 3 e 4 mostram o erro de treinamento para 1, 2, 10 e 30 elementos com uma única camada intermediária. As amostras de treinamento são mostradas na tabela 5.1, e o resultado de treinamento da rede para este caso está ilustrado na figura 5.1.

Tabela 5.1 Amostras de treinamento exemplo 1 (ou-exclusivo)

ENTRADA		SAÍDA
x_1	x_2	$f_1(x_1, x_2)$
0	0	0
1	0	1
0	1	1
1	1	0



Caso 2

As curvas 5,6,7 e 8 (figura 5.3) referem-se ao estado onde a rede é treinada admitindo-se uma tolerância de erro de 0.1 em relação aos padrões exatamente corretos $\{(0,0), (0,1), (1,0) \text{ e } (1,1)\}$, isto equivale a dizer que estamos tomando padrões em um disco de raio 0.1 (vide figura 5.2) e centros nos padrões $(0,0), (0,1), (1,0), (1,1)$ aleatoriamente, e com o número de elementos na camada intermediária igual ao referido no caso 1.

Na figura 5.3, as curvas 5, 6, 7 e 8 mostram o erro de treinamento para 1, 2, 10 e 30 elementos na camada intermediária. A diferença básica, neste caso, é que o mínimo global nunca é atingido, confirmando os resultados teóricos obtidos neste trabalho.

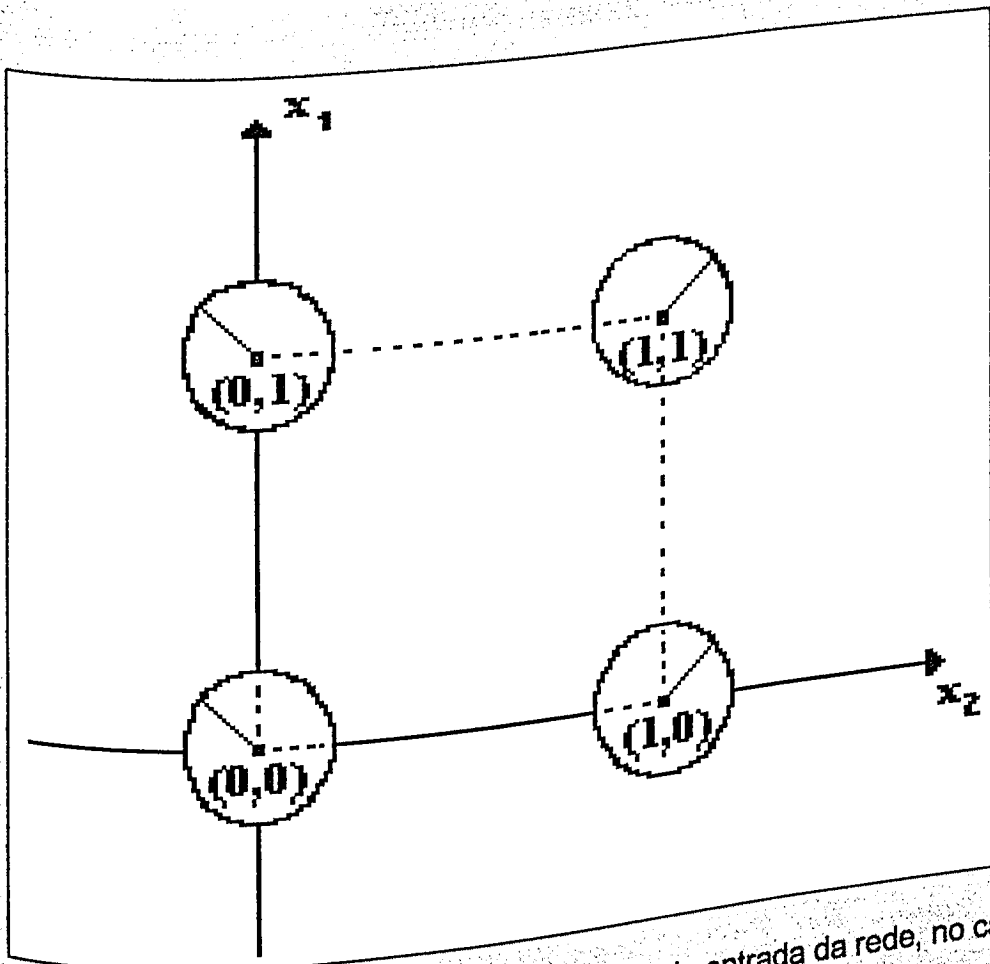
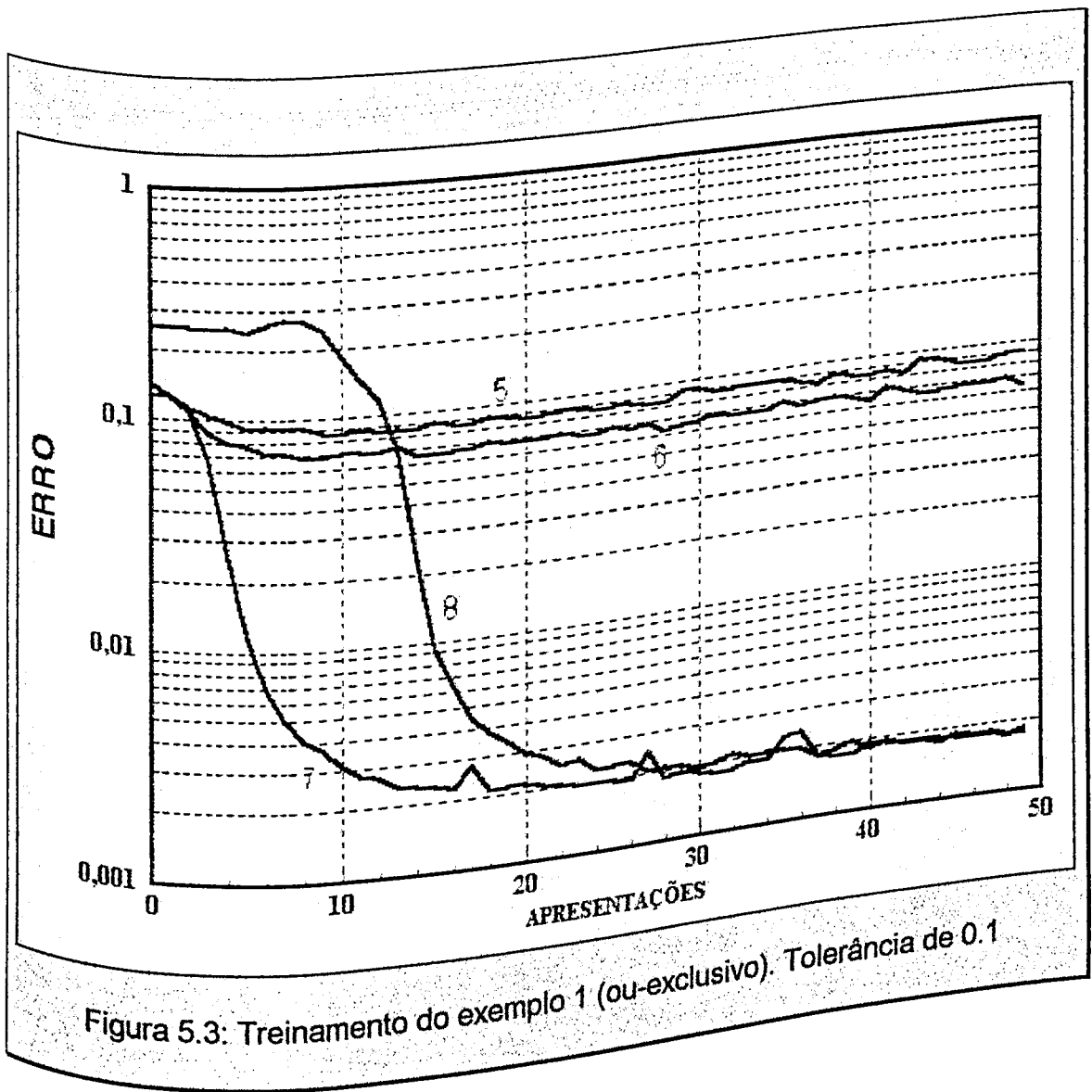


Figura 5.2: Representação geométrica dos padrões de entrada da rede, no caso 2.

A figura 5.3 ilustra a convergência da rede neural do sistema descrito no exemplo 1, caso 2.



Os resultados obtidos dos casos 1 e 2 descritos acima estão ilustrados na figura 5.4

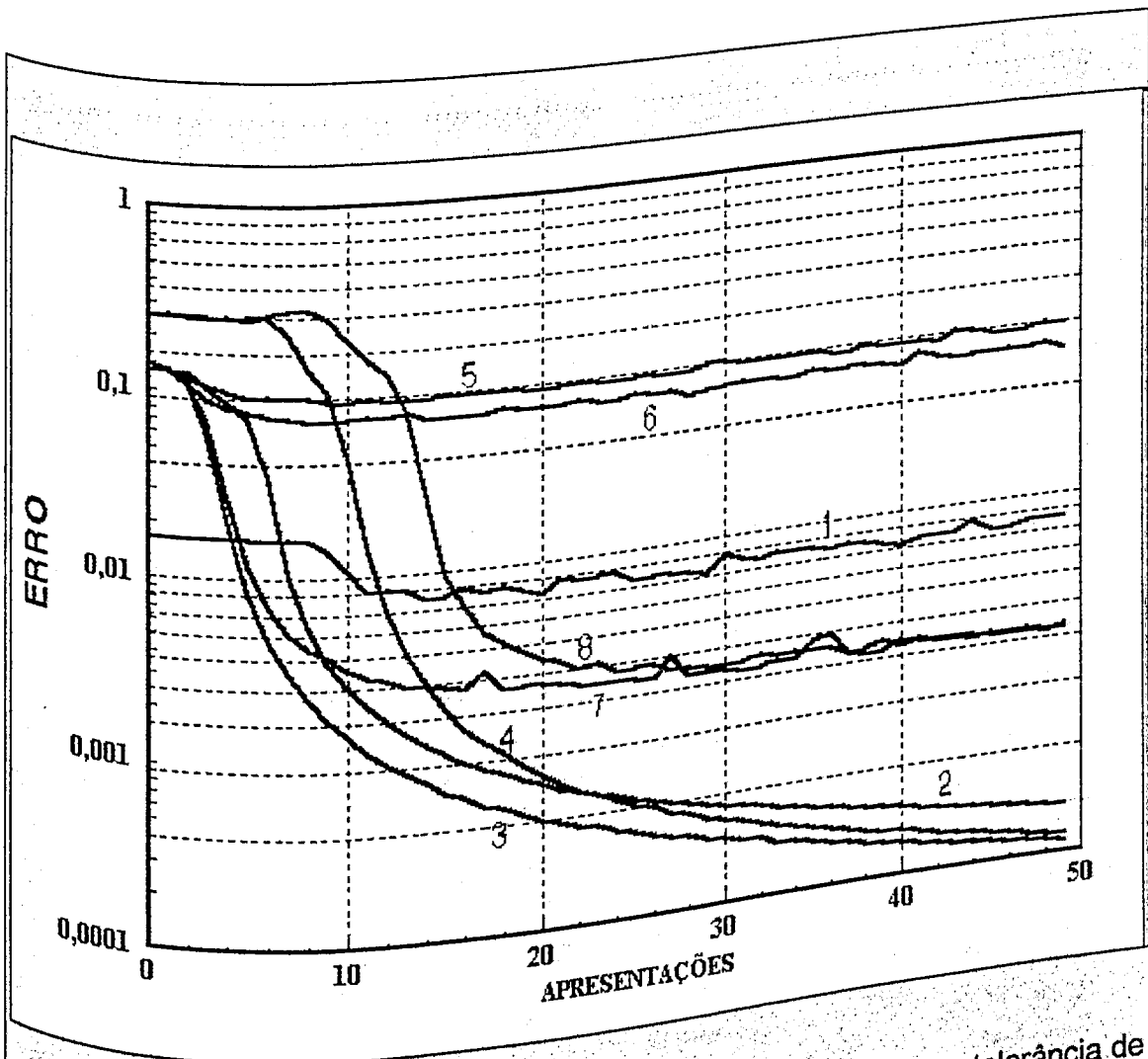


Figura 5.4: Treinamento do exemplo 1 (ou-exclusivo). Curvas 1,2,3,4 - tolerância de zero; curvas 5,6,7,8 - tolerância de 0.1

Analisando os resultados obtidos quando se tomam apenas os quatro padrões independentes $\{(0,0), (0,1), (1,0) \text{ e } (1,1)\}$, verifica-se que não é necessário um número muito grande de parâmetros de ajuste para ocasionar a convergência da rede. A admissão de uma tolerância para os padrões de entrada é equivalente a se

ter mais padrões distintos e, conseqüentemente, mais equações. Assim sendo verifica-se através da figura 5.4 (curvas 5 e 6) que, quando se toma um número pequeno de elementos na camada intermediária (poucos parâmetros de ajuste), não há convergência, ou seja, não se encontra solução para o sistema. Entretanto, aumentado-se o número de elementos da camada intermediária (curvas 7 e 8) observa-se uma leve tendência à convergência apesar desta nunca ocorrer. Isto acontece pois, ao se adotar a tolerância para os padrões cria-se um número infinito de amostras distintas e, portanto seria necessário um número infinito de parâmetros de ajuste para se atingir o erro nulo.

Exemplo 2

Considere a seguinte função $f_2(x_1, x_2, x_3) = x_1 \otimes x_2 \otimes x_3$ [15], $x_i \in \{0,1\}$, $i = 1, 2, 3$ e $x_1 \otimes x_2 \otimes x_3 \in \{0,1,0,9\}$, a operação acima está definida na tabela 5.2, geometricamente isto equivale a dizer que os padrões de treinamento (x_1^p, x_2^p, x_3^p) , onde $p = 1, 2, \dots, 8$ são os vértices do cubo de aresta de comprimento unitário, situado no primeiro octante de \mathbb{R}^3 . Os elementos de A_{tm} (conjunto de amostras de treinamento) usados para o treinamento estão mostrados na tabela 5.2 abaixo:

Tabela 5.2 Amostras de treinamento exemplo 2

(x_1, x_2, x_3)	$f_2(x_1, x_2, x_3)$	(x_1, x_2, x_3)	$f_2(x_1, x_2, x_3)$
$(0, 0, 0)$	0.1	$(1, 0, 0)$	0.9
$(0, 0, 1)$	0.9	$(1, 0, 1)$	0.1
$(0, 1, 0)$	0.9	$(1, 1, 0)$	0.1
$(0, 1, 1)$	0.1	$(1, 1, 1)$	0.9

Oito padrões são apresentados à rede. Observa-se que as curvas numeradas de 1 a 4 (figura 5.6) referem-se ao estado onde a rede é treinada com os padrões $(0, 0, 0)$, $(0, 0, 1)$, $(0, 1, 0)$, $(0, 1, 1)$, $(1, 0, 0)$, $(1, 0, 1)$, $(1, 1, 0)$, $(1, 1, 1)$ e com o número de elementos na camada intermediária iguais a 1, 2, 10 e 30, respectivamente.

A figura 5.5 abaixo representa os padrões de treinamento ilustrados na tabela 5.2 acima.

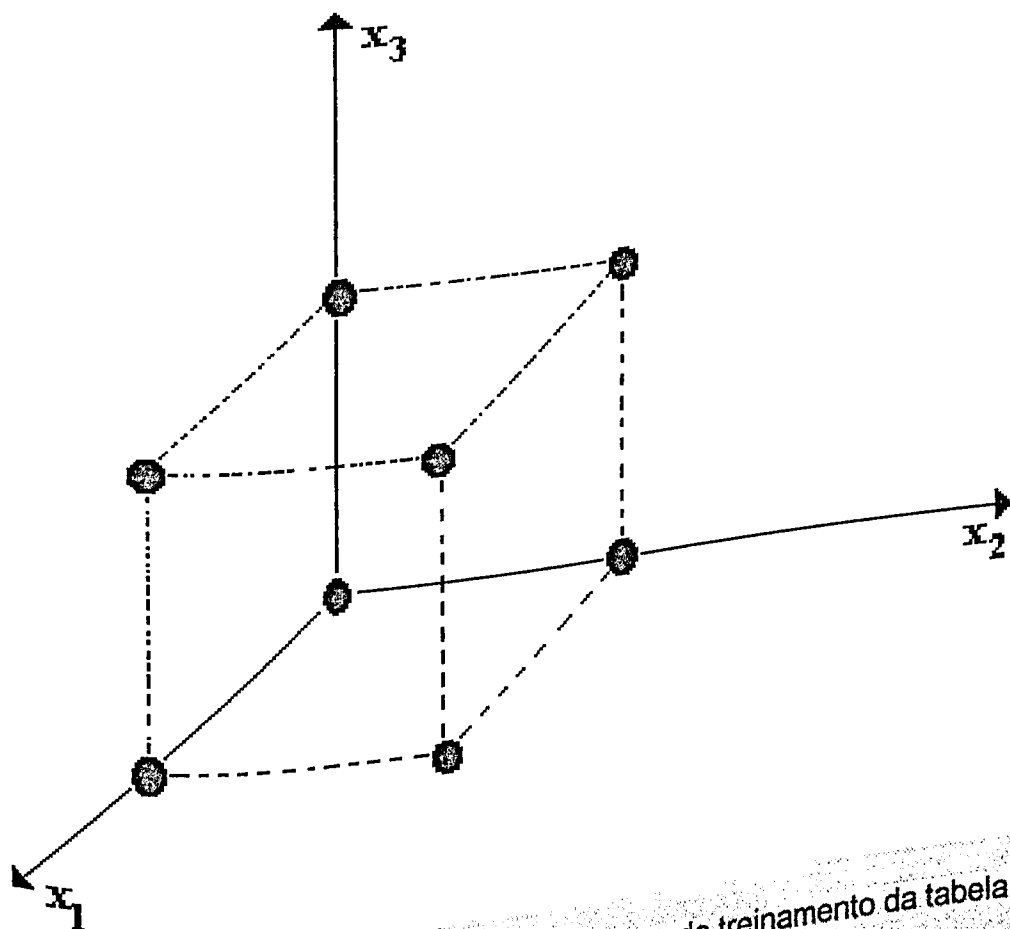
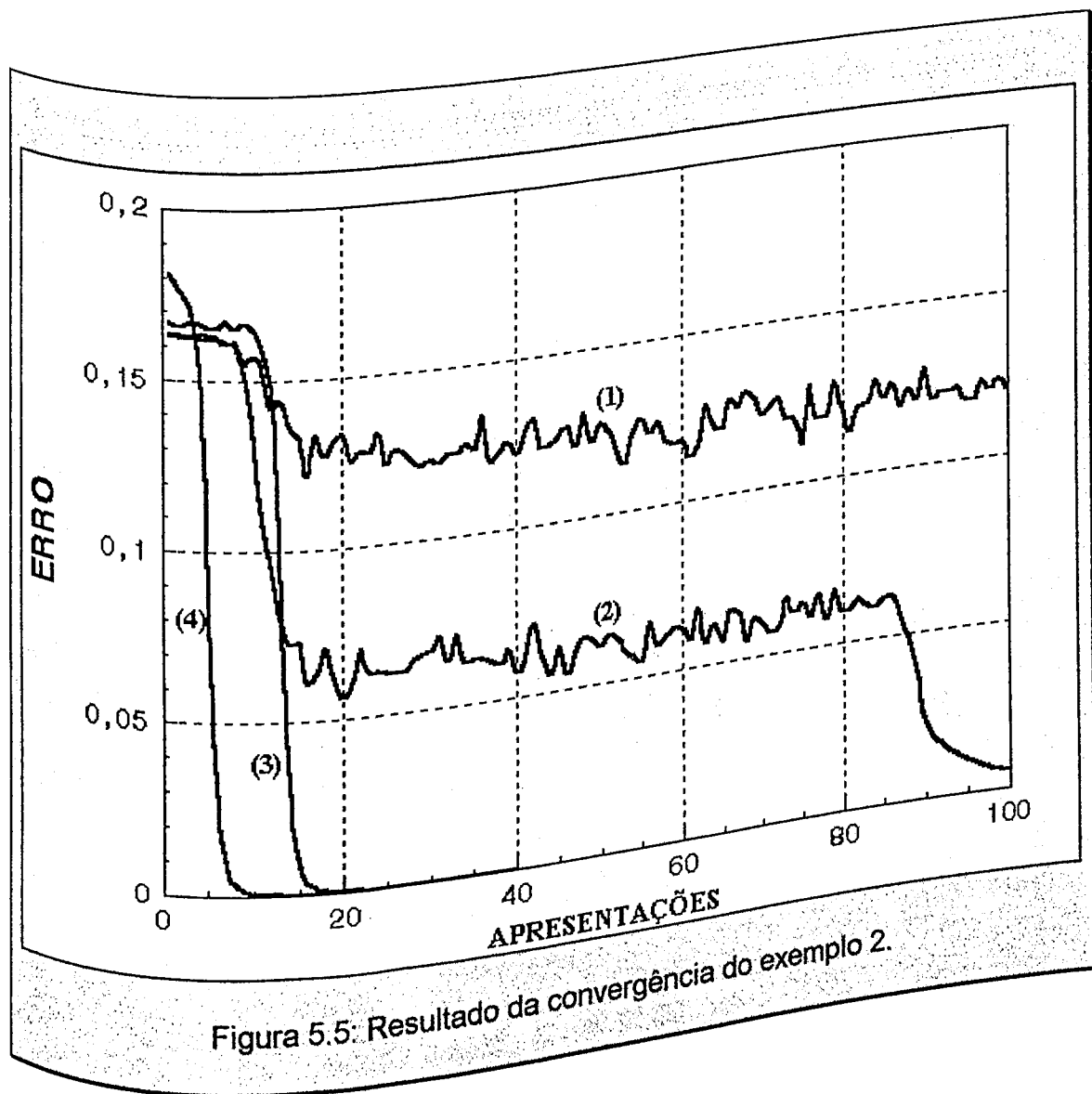


Figura 5.5 Representação geométrica dos padrões de treinamento da tabela 5.2

A figura 5.6 ilustra a convergência da rede neural do sistema descrito no exemplo 2.



Analisando-se os resultados obtidos na figura 5.5, observa-se que as curvas 2,3,4 convergem para o erro mínimo, o que não ocorre para a curva 1. Isto se explica pois, no caso da curva 1, a rede é treinada com um único elemento na

camada intermediária e, conseqüentemente com seis parâmetros de ajuste, Portanto, o número de parâmetros de ajuste é menor do que o número de amostras distintas tomadas para treinamento. No caso das curvas 2, 3 e 4, a rede é treinada com 2, 10 e 30 elementos na camada intermediária e, consecutivamente com o número de parâmetros de ajuste maior que o número de amostras distintas usadas para o treinamento da rede.

5.2 CONCLUSÃO

Os resultados obtidos das simulações dos exemplos 1 e 2, apresentados anteriormente, oferecem uma comprovação prática do teorema 4.3.1 e corolários decorrentes do mesmo. Embora este assunto tenha sido bastante discutido pelos pesquisadores da área, os resultados obtidos neste trabalho oferecem um novo caminho para dimensionar redes neurais. Oferece também uma visão clara do fenômeno da convergência de redes neurais multi-camadas, por tratar do assunto como um problema de solução de um sistema de equações algébricas não-lineares. Observa-se ainda que este método, é muito bem aplicado quando as amostras de treinamento são conhecidas previamente. Ainda é motivo de pesquisa a execução de um pré-processamento no conjunto de dados, no sentido de eliminar padrões múltiplos ou de pouca representatibilidade.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Rumelhart, D. & McClelland, J.; "Parallel Distributed Processing", Vol. 1 e 2, MIT Press, Cambridge, Massachusetts, 4a. edição, 1987.
- [2] Lippman, R.; "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, Abril 1987, pp 4-22.
- [3] Haaser, N. & Sullivan, J.; "Real Analysis", New York Van Nostrand Reinhold, 1971.
- [4] Wang, J. & Malakooti, B.; "On training of Artificial Neural Networks", Proc. of IEEE/INNS International Joint Conf. On Neural Networks, Vol II, 387-393, Washington, Dc, junho 1989.
- [5] Funakashi, K.; "On the Approximate Realization of Continuous Mappings By Neural Networks", Neural Networks, 2(3), 183-192, 1989.
- [6] Baum, E. & Hausler, D.; "What Size net Gives Valid Generalization?", Advances in Information Processing Systems 1, Morgan Kaufmann Publishers, San Mateo Ca 94403, pp. 81, 1989.
- [7] Hornik, K.; & Stinchcombe, M. & White, H.; "Multilayer Feedforward Networks are Universal Approximators", Springer - Verlag, 1989, 2a. edição.
- [8] Lima, Elon Lages, 1929; "Curso de Análise", Vol. 2, Rio de Janeiro, Instituto de Matemática Pura e Aplicada, CNPq, edição, 1981.

- [9] Buck, R; "Advanced Calculus", McGraw-Hill Book Company, 1978.
- [10] Cotter, E. Neil, "The Stone-Weierstrass Theorem and Its Application to Neural Networks", IEEE Transactions on Neural networks, Vol I, No. 4, pp 290-295, December 1990.
- [11] Hoffman, Kenneth [e] Kunze, Ray; "Algebra Linear". 2^a ed. - Rio de Janeiro: LTC, 1979.
- [12] Kurková, Vera, "Kolmogorov's Theorem and Multilayer Neural Networks", Neural Networks, Vol. 5, pp. 501-506, 1992.
- [13] Lima, Elon Lages, "Espaços Métricos", Rio de Janeiro, Instituto de Matemática Pura e Aplicada, CNPq, 1977.
- [14] Simpson, Patrick K., "Artificial Neural Systems", Pergamon Press, 1990.
- [15] Wang, Jun & Malakooti B., "Analysis and Synthesis of Artificial Neural Networks for Modeling of Complex Systems", in Progress in Neural Networks, Vol. 2, O. M. Omidvar (ed), Norwood, NJ: Ablex Publishing, 1990.
- [16] Lorentz, G. G., (1976) "The thirteenth problem of Hilbert". In F. E. Browder (Ed), Proceedings of Symposia in Pure Mathematics (Vol 28, p.p. 419-430). Providence, RI: American Mathematical Society.
- [17] Kolmogorov, A. N. (1957). "On the representations continuous of many variables by superpositions of continuous functions of one variable and addition". Doklady Akademii Nauk. USSR, 114, 679-681.
- [18] Teixeira, Edilberto P., "Controle de Sistemas não-lineares usando Redes Neurais". Tese de Doutorado, UNICAMP, 1991.

[19] Kuo, Benjamin C., "Digital Control Systems", (HRW series in electrical and computer engineering) Part I, 1980.

[20] Teixeira, Edilberto P. & Belonsi, Marcelo H., "Using the Implicit Function Theorem for Dimensioning Multi-Layer Neural Networks", International Fuzzy Systems Association, Vol II, pag. 655-658, São Paulo - Brazil, 1995

Anexo

Apresenta-se neste anexo os valores obtidos no treinamento da rede neural (algoritmo backpropagation [14]), assim como vistos nos gráficos do capítulo 5.

Tabela A: Erro de treinamento da RNA para o exemplo 1, caso1.

CRUVAS APRESENTAÇÕES	ERRO			
	1	2	3	4
1	0.1261936236	0.1277155584	0.1370808366	0.2664631436
2	0.1268943822	0.1262347955	0.1224390360	0.2544544168
3	0.1219945670	0.1236307070	0.0959560956	0.2474388037
4	0.1119127786	0.1127830837	0.0596379412	0.2519168958
5	0.0933781398	0.0964616019	0.0238220980	0.2573547582
6	0.0919275113	0.0914814818	0.0097952399	0.2368365631
7	0.0917407955	0.0956387038	0.0053478836	0.1427416028
8	0.0851311367	0.0873507668	0.0034189536	0.1213754371
9	0.0878988104	0.0850760584	0.0025825929	0.0772477479
10	0.0861824198	0.0887384040	0.0019543518	0.0273791591

CRUVAS APRESENTAÇÕES	ERRO			
	1	2	3	4
11	0.0874090368	0.0905803807	0.0015232692	0.0092557246
12	0.0863388631	0.0851131598	0.0012940802	0.0044618173
13	0.0861003560	0.0868709181	0.0011092070	0.0028547319
14	0.0873577940	0.0831368303	0.0009458716	0.0020376410
15	0.0872078769	0.0819828823	0.0008316573	0.0015639153
16	0.0832416148	0.0329477416	0.0007279110	0.0012464429
17	0.0842787162	0.0090527219	0.0006631394	0.0010202545
18	0.0833014214	0.0045329128	0.0006106885	0.0008714077
19	0.0805992626	0.0029550281	0.0005532016	0.0007321158
20	0.0833061329	0.0021630216	0.0004889883	0.0006351802
21	0.0897527348	0.0017155666	0.0004603378	0.0005778254
22	0.0822678284	0.0014002312	0.0004261674	0.0005148375
23	0.0811668170	0.0011850887	0.0004041057	0.0004640389
24	0.0871287116	0.0010248940	0.0003835961	0.0004246336
25	0.0871979487	0.0009027572	0.0003576413	0.0003919351
26	0.0819318459	0.0008012137	0.0003343119	0.0003658859
27	0.0874454814	0.0007258741	0.0003114367	0.0003343009
28	0.0828443212	0.0006686666	0.0002945721	0.0003129205
29	0.0897646298	0.0006073646	0.0002830241	0.0002975888
30	0.0885377938	0.0005672379	0.0002662368	0.0002783003
31	0.0798015512	0.0005275524	0.0002585620	0.0002598048
32	0.0815265226	0.0004940986	0.0002362423	0.0002475391
33	0.0831295998	0.0004631797	0.0002362801	0.0002343232
34	0.0823835024	0.0004357790	0.0002251669	0.0002149329
35	0.0883825786	0.0004133638	0.0002129316	0.0002117177

CRUVAS APRESENTAÇÕES	ERRO			
	1	2	3	4
36	0.0875042609	0.0003902456	0.0002072565	0.0002021284
37	0.0866370364	0.0003689877	0.0002006595	0.0001884060
38	0.0876367528	0.0003517245	0.0001922802	0.0001825455
39	0.0839360525	0.0003378491	0.0001852938	0.0001757697
40	0.0813704052	0.0003224240	0.0001802355	0.0001679131
41	0.0819861070	0.0003081009	0.0001744442	0.0001639000
42	0.0869383993	0.0002951478	0.0001629403	0.0001524272
43	0.0806120517	0.0002851564	0.0001603608	0.0001481399
44	0.0851611773	0.0002755329	0.0001591079	0.0001464642
45	0.0840554848	0.0002641685	0.0001501619	0.0001424016
46	0.0871860318	0.0002558801	0.0001443395	0.0001361811
47	0.0856518337	0.0002471003	0.0001428851	0.0001316135
48	0.0866107863	0.0002401178	0.0001380828	0.0001245334
49	0.0897591733	0.0002315468	0.0001359963	0.0001216455
50	0.0891109873	0.0002235176	0.0001310423	0.0001187233

Tabela B: Erro de treinamento da RNA para o exemplo 1, caso 2.

CRUVAS APRESENTAÇÕES	ERRO			
	5	6	7	8
1	0.1272635042	0.1291311987	0.1454398984	0.2434810753
2	0.1275322153	0.1284837802	0.1270377018	0.2589783965
3	0.1258972191	0.1269940948	0.1112028165	0.2604775760
4	0.1169690506	0.1263400220	0.0739912542	0.2499755472
5	0.1052567536	0.1239564318	0.0346821144	0.2449727365
6	0.0957076956	0.1117104171	0.0133554586	0.2629650959
7	0.0902570566	0.0890039165	0.0065952791	0.2654577853
8	0.0886080598	0.0793953179	0.0040969597	0.2384480961
9	0.0850500435	0.0694984024	0.0028717965	0.2519276266
10	0.0926105750	0.0663585379	0.0021185030	0.2523961807
11	0.0903792460	0.0683568118	0.0015691638	0.2612642550
12	0.0868363369	0.0651802927	0.0013367834	0.1999030865
13	0.0829240128	0.0651802927	0.0011131157	0.1391901883
14	0.0896605189	0.0678541133	0.0009173219	0.1243326591
15	0.0917453129	0.0642356548	0.0012582452	0.0905426704
16	0.0857445610	0.0698765457	0.0007361433	0.0405893792
17	0.0834026902	0.0697187924	0.0006456797	0.0142033926
18	0.0845020403	0.0653177774	0.0006456797	0.0066661669
19	0.0875070787	0.0668663450	0.0005944215	0.0040300458
20	0.0856819072	0.0628894306	0.0005266115	0.0027806836
21	0.0855356600	0.0663933320	0.0004770852	0.0020389555
		0.0640655304	0.0004637519	

CRUVAS APRESENTAÇÕES	ERRO			
	5	6	7	8
22	0.0830591160	0.0633623357	0.0004233718	0.0015909688
23	0.0835329653	0.0664791811	0.0003992535	0.0012829146
24	0.0838468586	0.0647737471	0.0003589444	0.0010763484
25	0.0848369076	0.0676636152	0.0003494656	0.0009215987
26	0.0877843641	0.0679298788	0.0003202830	0.0007940370
27	0.0794785648	0.0674286504	0.0003179859	0.0006905332
28	0.0844855228	0.0671628630	0.0002903884	0.0006153791
29	0.0821637668	0.0624727506	0.0002700266	0.0005217026
30	0.0839512311	0.0681106602	0.0002585523	0.0005377475
31	0.0820606370	0.0641726977	0.0002560908	0.0004491359
32	0.0816320167	0.0628389252	0.0007178160	0.0004362841
33	0.0887706874	0.0636385058	0.0002258638	0.0004046597
34	0.0888624683	0.0638278501	0.0002174107	0.0003584748
35	0.0862001681	0.0668495466	0.0002118923	0.0003480253
36	0.0847819246	0.0633874505	0.0001992379	0.0003208330
37	0.0857989583	0.0639558840	0.0001922994	0.0002970532
38	0.0885092345	0.0675842754	0.0001883878	0.0002965260
39	0.0872186039	0.0644218181	0.0001806785	0.0002624206
40	0.0831674772	0.0647378303	0.0001730767	0.0002577744
41	0.0862180240	0.0667360175	0.0001631075	0.0002315778
42	0.0819577810	0.0614344133	0.0001633648	0.0002318203
43	0.0913816535	0.0649262506	0.0001526294	0.0002208019
44	0.0858689991	0.0617944641	0.0001506669	0.0002182953

		ERRO			
		5	6	7	8
CRUVAS	APRESENTAÇÕES				
	45	0.0879660842	0.0640768546	0.0001476500	0.0002087657
	46	0.0797193041	0.0650779284	0.0001434574	0.0001938357
	47	0.0903310079	0.0633851749	0.0001322702	0.0001778850
	48	0.0906176043	0.0673914390	0.0001325547	0.0001773135
	49	0.0818968065	0.0694529361	0.0001255302	0.0001752865
	50	0.0821798295	0.0632277305	0.0001235085	0.0001672076

Tabela C: Erro de treinamento da RNA para o exemplo 2.

		ERRO			
		1	2	3	4
CRUVAS	APRESENTAÇÕES				
	1	0,1626612577	0,1633180827	0,1667235273	0,1807092310
	2	0,1627355352	0,1636294927	0,1655927826	0,1765486105
	3	0,1622600931	0,1627226052	0,1665859625	0,1722540317
	4	0,1626810171	0,1635724290	0,1669653520	0,1578991836
	5	0,1615605828	0,1636121891	0,1655042297	0,0994404688
	6	0,1627056245	0,1621267165	0,1653456427	0,0316901055
	7	0,1605573217	0,1609759723	0,1672384279	0,0085007281
	8	0,1610183561	0,1605491212	0,1654300145	0,0029905831

CRUVAS APRESENTAÇÕES	ERRO			
	1	2	3	4
9	0,1543620047	0,1535841008	0,1662498343	0,0012875014
10	0,1568720568	0,1368645105	0,1652731899	0,0006157568
11	0,1541777659	0,1097067471	0,1593949919	0,0003125432
12	0,1418451505	0,0942471357	0,1467987230	0,0001716827
13	0,1438806669	0,0812224033	0,0946786153	0,0000924410
14	0,1364944622	0,0726333008	0,0317062412	0,0000529192
15	0,1333409073	0,0729952347	0,0090342710	0,0000312351
16	0,1214736168	0,0610911485	0,0033213834	0,0000180143
17	0,1331419358	0,0634480463	0,0015121334	0,0000104686
18	0,1269505166	0,0711163715	0,0007266815	0,0000065902
19	0,1312804263	0,0616380878	0,0003999818	0,0000040391
20	0,1331377529	0,0564516243	0,0002230796	0,0000024082
21	0,1248744808	0,0597141483	0,0001337640	0,0000015019
22	0,1276431139	0,0695871644	0,0000769613	0,0000009698
23	0,1272111419	0,0629384596	0,0000527468	0,0000005671
24	0,1327863375	0,0629016654	0,0000318916	0,0000003592
25	0,1233011166	0,0625411433	0,0000201517	0,0000002204
26	0,1271062922	0,0622617488	0,0000138344	0,0000001393
27	0,1260681339	0,0617705385	0,0000093519	0,0000000854
28	0,1236609431	0,0650519233	0,0000059933	
29	0,1215479596	0,0664807859	0,0000040214	
30	0,1231252936	0,0672245376	0,0000027584	
31	0,1219680773	0,0711694766	0,0000017892	

CRUVAS APRESENTAÇÕES	ERRO			
	1	2	3	4
32	0,1241499966	0,0634132992	0,0000013426	
33	0,1240684543	0,0708123255	0,0000010411	
34	0,1264966541	0,0625017397	0,0000006481	
35	0,1249539896	0,0629403419	0,0000004987	
36	0,1340934452	0,0632190195	0,0000003273	
37	0,1205248249	0,0617431677	0,0000002504	
38	0,1228789486	0,0606177294	0,0000001908	
39	0,1252806849	0,0646252843	0,0000001355	
40	0,1219726811	0,0573124939	0,0000000935	
41	0,1279825680	0,0615612014		
42	0,1313347892	0,0707317484		
43	0,1226268206	0,0614757182		
44	0,1232105837	0,0554667341		
45	0,1255861296	0,0632526902		
46	0,1274390559	0,0546224646		
47	0,1213802471	0,0605917115		
48	0,1306821660	0,0647233384		
49	0,1221505908	0,0637530743		
50	0,1268898470	0,0614021257		
51	0,1250721778	0,0643943038		
52	0,1188341060	0,0628667110		
53	0,1139754272	0,0586763076		
54	0,1235240292	0,0565172701		

CRUVAS APRESENTAÇÕES	ERRO			
	1	2	3	4
55	0,1267464724	0,0547431951		
56	0,1217384557	0,0660705157		
57	0,1245213661	0,0588528878		
58	0,1194600046	0,0611013689		
59	0,1177282389	0,0629060205		
60	0,1180041400	0,0628312637		
61	0,1129472320	0,0588406489		
62	0,1151794804	0,0678284393		
63	0,1274878369	0,0571399296		
64	0,1210738138	0,0619368952		
65	0,1194856144	0,0565707489		
66	0,1283041178	0,0655785866		
67	0,1254224844	0,0649156125		
68	0,1300501683	0,0555326911		
69	0,1278225241	0,0618456534		
70	0,1221453986	0,0602320420		
71	0,1244460244	0,0569817862		
72	0,1267030848	0,0579325814		
73	0,1197789364	0,0662856101		
74	0,1195365061	0,0617657024		
75	0,1115402725	0,0644303750		
76	0,1285922507	0,0596560604		
77	0,1186374711	0,0659241127		

CRUVAS APRESENTAÇÕES	ERRO			
	1	2	3	4
78	0,1188711220	0,0581855853		
79	0,1288027171	0,0665169378		
80	0,1203987719	0,0597650049		
81	0,1129361882	0,0605300358		
82	0,1203281315	0,0629337221		
83	0,1209217563	0,0609454078		
84	0,1277687002	0,0597937513		
85	0,1221624235	0,0619727502		
86	0,1261794041	0,0627670905		
87	0,1188454172	0,0558136567		
88	0,1234384689	0,0485202845		
89	0,1204826771	0,0389944503		
90	0,1293748472	0,0231276047		
91	0,1198918513	0,0179265113		
92	0,1216117788	0,0155437940		
93	0,1212496683	0,0132306560		
94	0,1227520003	0,0114483990		
95	0,1184152752	0,0099920461		
96	0,1179419232	0,0087989260		
97	0,1234554433	0,0074958058		
98	0,1211388518	0,0064205765		
99	0,1237789042	0,0059976975		
100	0,1204039620	0,0052507743		

N. REGISTRO 00300/96 ex. 1



Universidade Federal de Uberlândia

DATA DA DEVOLUÇÃO

Esta obra deve ser devolvida na última data carimbada

ESTA OBRA NÃO PODE SER
RETIRADA DA BIBLIOTECA

PROEPE 175



Universidade Federal de Uberlândia

00300/96 ex. 1

Belonsi, Marcelo Henrique

O uso do teorema da função implícita no dimensionamento de redes neurais multi-camadas
681.3:007.52/B452u/TES/FU ex.1
(300/96)

ESTA OBRA NÃO PODE SER
RETIRADA DA BIBLIOTECA

PROEPE 086