

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA QUÍMICA
ÁREA DE CONCENTRAÇÃO: SISTEMAS DE PROCESSOS QUÍMICOS E
INFORMÁTICA

**MODELAGEM MATEMÁTICA DA EXTRAÇÃO DO
CITOCROMO B5 VIA REDES NEURAIS**

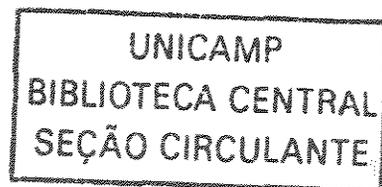
Autor : Gilvan Anderson Fischer

Orientador : Prof. Dr. Elias Basile Tambourgi

Co-orientadora: Prof^ª. Dra. Ana M. Frattini Fileti

Dissertação de Mestrado apresentada
a Faculdade de Engenharia Química
como parte dos requisitos exigidos
para obtenção do título de Mestre em
Engenharia Química.

SETEMBRO / 2002
CAMPINAS – SÃO PAULO



UNIDADE	BC
Nº CHAMADA	UNICAMP
	F 523 m
V	EX
TOMBO BC/	52063
PROC.	16-124103
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	78 11,00
DATA	12/02/03
Nº CPD	

CM00179039-9

B/B 10 276180

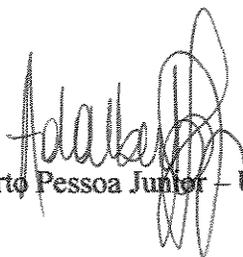
FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

<p>F522m F523m</p>	<p>Fischer, Gilvan Anderson Modelagem matemática da extração do citocromo B5 via redes neurais / Gilvan Anderson Fischer.-- Campinas, SP: [s.n.], 2002.</p> <p>Orientadores: Elias Basile Tambourgi e Ana Maria Frattini Fileti. Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Química.</p> <p>1. Simulação (computadores). 2. Extração (Química). 3. Inteligência artificial. 4. Redes neurais (Computação). I. Tambourgi, Elias Basile. II. Fileti, Ana Maria Frattini. III. Universidade Estadual de Campinas. Faculdade de Engenharia Química. IV. Título.</p>
-----------------------------------	--

Dissertação de Mestrado defendida por Gilvan Anderson Fischer em 26 de setembro de 2002 e aprovada pela banca constituída pelos seguintes doutores:



Elias Basile Tambourgi – UNICAMP

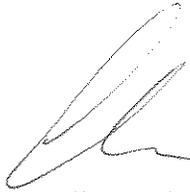


Adalberto Pessoa Junior – USP



Ana Paula Brescancini Rabelo – pesquisadora FAPESP

Esta versão corresponde a final da Dissertação de Mestrado , defendida por
Gilvan Anderson Fischer , em 26 de setembro de 2002.

A handwritten signature in black ink, consisting of several fluid, overlapping strokes that form a stylized representation of the name Elias Basile Tambourgi.

Prof Dr Elias Basile Tambourgi - orientador

Agradecimentos

A Deus.

Ao professor Elias Basile Tambourgi, pela orientação, amizade e incentivo que estiveram presentes durante o desenvolvimento do projeto.

A professora Ana M. Frattini Fileti pela co-orientação, amizade, e pela oportunidade do estágio docência, muito importantes durante a execução do projeto.

Aos amigos Romi, Newton, Itaciara, Virgínia e Viktor pelo companheirismo, amizade e apoio durante este trabalho.

Aos amigos Jones e Rafael pelas dicas e trocas de experiências durante o trabalho.

Aos colegas de curso Priscila, Gaston, Carlos e Luciane pelo coleguismo durante o curso.

Aos professores e funcionários da Faculdade de Engenharia Química pelo apoio e amizade dispensados durante a realização do trabalho.

A Ana Paula Rabelo por disponibilizar os dados e pelas dicas sobre o trabalho.

A Capes pelo suporte financeiro.

À minha família pelo incentivo, apoio, amizade, interesse e paciência que foram a mim dispensados.

A Vera minha companheira pelo amor, amizade, apoio e compreensão.

A todos os amigos que direta ou indiretamente colaboraram na realização deste trabalho.

Resumo

A modelagem da extração líquido-líquido pode ser feita através de abordagens fenomenológicas ou empíricas. Para a modelagem fenomenológica é necessária a compreensão dos fenômenos físicos e químicos, e nem todos são conhecidos. Na abordagem empírica não é necessário o conhecimento dos fenômenos envolvidos. Uma das abordagens empíricas é o uso de redes neurais artificiais. Estes modelos têm a habilidade de ‘aprender’ comportamentos complexos do sistema físico. O aprendizado é um procedimento pelo qual a rede aproxima uma função mapeando um conjunto de dados de entrada-saída. As redes neurais podem ser usadas como ‘caixas pretas’, não é necessário nenhum conhecimento sobre o comportamento do processo.

Neste trabalho foram desenvolvidos modelos para a extração do citocromo b5 em operação descontínua e em operação contínua. Em operação descontínua a entrada foi a massa molar do PEG, e as viscosidades das fases PEG e sal, a saída foi o coeficiente de partição, e em operação contínua a entrada usada foi o coeficiente de partição, frequência de pulsação e vazão total das fases, sendo a saída a porcentagem de extração e a fração de retenção da fase dispersa. Os algoritmos usados para o treinamento das redes foram: Levenberg-Marquardt, Parada Antecipada e Regularização Bayesiana. A entrada e saída foram normalizadas entre 0,1 e 0,9 e as funções de ativação do tipo sigmoidal.

Verificou-se que dentre os modelos desenvolvidos o que teve melhor ajuste foi a rede treinada com o algoritmo Regularização Bayesiana, mas o algoritmo parada antecipada também obteve bons ajustes. Os erros dos modelos ficaram dentro da faixa esperada. Assim, o principal objetivo deste trabalho, que era a modelagem da extração do citocromo b5, com erro dentro da faixa experimental, foi alcançada com sucesso.

Abstract

The liquid-liquid extraction modeling can be done through phenomenologic or empiric approach . For phenomenologic modeling is necessary understanding of physic and chemical phenomenon, and all neither are known. In the empiric approach is not necessary the knowledge about the involved phenomenon. One of empiric aproach is the use of artificial neural networks. This models have the habilit of 'learn' complex physic systems behaviour. The learnt is the behaviour which the network approaches function, maping an input-output data set. Neural networks can be used how 'black boxes' is not necessary knowlege about process behaviour.

In this work was developed model for cytochrome b5 extraction in descontinuos operation and continuous operation. The descontinuos input was: PEG molecular weight, PEG and salt viscosities, the output was partition coefficient, in the continuous operation the input was: partition coefficient, pulsation frequency and phases overall flowrate, the output extraction percentage and disperse phase holdup. The algoritms used for training was: Levemberg-Marquardt, Early Stoping and Bayesian Regularization. The input-output was scaled into the range 0,1-0,9, and activation function sigmoidal type.

It was observed the models developed, the better adjust was the network trained with Bayesian Regularization algoritm, but the Early Stoping algoritm had hight adjust too. The model errors stayed into the range expected. Then, the main objetive of this work, the modeling cytochrome b5 extraction, with error into experimental range, was obtained with success.

ÍNDICE

ÍNDICE DE FIGURAS	2
ÍNDICE DE TABELAS	5
NOMENCLATURA	7
ABREVIATURAS	10
1.0 Introdução	11
2.0 Revisão de Literatura e Fundamentos Teóricos	13
2.1 Proteínas	13
2.1.1 Citocromo b5	14
2.2 Etapas de Separação em Processos Biotecnológicos	14
2.2.1 Extração Líquido-Líquido	16
2.3 Sistemas de Duas Fases Aquosas	17
2.3.1 Partição de Proteínas em Sistemas de Duas Fases Aquosas	20
2.4 Equipamentos Usados nos Processos de Extração	22
2.4.1 Coluna Agitada por Campânulas Pulsantes	22
2.4.2 Descrição do Equipamento	24
2.5 Extração do Citocromo b5 em Operação Descontínua	25
2.6 Eficiência de Extração	26
2.7 Fração de Retenção da Fase Dispersa	27
2.8 Redes Neurais Artificiais	28
2.8.1 O Modelo Computacional das RNA's	29
2.8.2 O Neurônio Artificial	30
2.8.3 Funções de Ativação	32
2.8.4 Topologia de uma Rede Neural Artificial	34
2.9 Treinamento da RNA	35
2.9.1 Dinâmica de treinamento	35
2.9.2 Tipos de Aprendizado	36
2.9.3 Alguns Tipos de Aprendizado	37
2.9.3.1 Aprendizado por Retropropagação ("Backpropagation")	39
2.9.3.2 Treinamento pela Regra de Delta Generalizado	42
2.9.3.3 Algoritmo de Treinamento Levenberg-Marquardt	46
2.10 Generalização	47
2.10.1 Regularização	47
2.10.2 Parada Antecipada	52
2.11 Uso de Redes Neurais em Extração Líquido-Líquido	53
3.0 Materiais e Métodos	54
3.1 Software	54
3.2 Algoritmos	54
4.0 Resultados e Discussão	54
4.1 Dados	54
4.2 Operação Descontínua	54
4.3 Operação Contínua	55
4.4 Normalização	56
4.5 Função de Ativação	57
4.6 Análise do Treinamento	57

4.7	Divisão do Conjunto de dados.....	57
4.7.1	Descontínuo.....	57
4.7.2	Contínuo.....	57
4.8	Treinamento.....	58
4.8.1	Operação Descontínua.....	58
4.8.1.1	Seleção das Variáveis.....	58
4.8.1.2	Variáveis.....	58
4.8.1.3	Treinamento para Seleção das Variáveis.....	59
4.8.1.4	Treinamento do Modelo.....	86
4.8.1.4.1	Treinamento com dados com variação na entrada.....	86
4.8.1.4.1.1	Algoritmo Levenberg-Marquardt.....	86
4.8.1.4.1.2	Algoritmo Parada Antecipada:.....	89
4.8.1.4.1.3	Algoritmo Regularização Bayesiana:.....	92
4.8.1.4.2	Treinamento com dados sem variação na entrada.....	95
4.8.1.4.2.1	Algoritmo Levenberg-Marquardt:.....	95
4.8.1.4.2.2	Algoritmo Parada Antecipada:.....	98
4.8.1.4.2.3	Algoritmo Regularização Bayesiana:.....	101
4.8.2	Operação Contínua.....	104
4.8.2.1	Seleção das Variáveis.....	104
4.8.2.2	Variáveis.....	104
4.8.2.3	Treinamento para Seleção das Variáveis.....	106
4.8.2.4	Treinamento do Modelo.....	114
4.8.2.4.1	Treinamento com dados com variação na entrada.....	115
4.8.2.4.1.1	Algoritmo Levenberg-Marquardt:.....	115
4.8.2.4.1.2	Algoritmo Parada Antecipada:.....	117
4.8.2.4.1.3	Algoritmo Regularização Bayesiana:.....	120
4.8.2.4.2	Treinamento com dados sem variação na entrada.....	122
4.8.2.4.2.1	Algoritmo Levenberg-Marquardt:.....	122
4.8.2.4.2.2	Algoritmo Parada Antecipada:.....	125
4.8.2.4.2.3	Algoritmo Regularização Bayesiana:.....	128
5.0	Conclusões e Sugestões.....	131
4.9	Conclusões.....	131
4.10	Sugestões.....	132
6.0	Referências Bibliográficas.....	133

ÍNDICE DE FIGURAS

Figura 2.1 – Agitação através de campânulas pulsantes numa coluna de extração líquido-líquido pulsada. (a) campânula parada durante o intervalo entre pulsos. (b) movimento descendente durante o pulso.....	23
Figura 2.2 – Esquema da micro coluna de campânulas pulsantes.....	24
Figura 2.3 - RNA e suas interconexões.....	29
Figura 2.4 - Modelo de um Neurônio Artificial.....	31
Figura 2.5 - Tipos de conexões entre os neurônios:.....	34
Figura 2.6 - Conexões por Alimentação Direta e Retro-Alimentação.....	34
Figura 2.7 – RNA do tipo Perceptron com três camadas.....	39

Figura 2.8 – Gráfico do treinamento com algoritmo Parada Antecipada.....	52
Figura 4.1- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.....	59
Figura 4.2 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.....	60
Figura 4.3 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.....	61
Figura 4.4- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.....	62
Figura 4.5- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.....	63
Figura 4.6- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.....	64
Figura 4.7- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.....	65
Figura 4.8- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.....	66
Figura 4.9 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.....	67
Figura 4.10- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.....	68
Figura 4.11 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.....	69
Figura 4.12 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.....	70
Figura 4.13- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.....	71
Figura 4.14- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.....	72
Figura 4.14- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.....	73
Figura 4.15- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.....	74
Figura 4.16- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.....	75
Figura 4.17- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.....	76
Figura 4.18- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.....	77
Figura 4.19- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.....	78
Figura 4.20- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.....	79
Figura 4.21- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.....	80
Figura 4.22- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.....	81

Figura 4.23- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.	83
Figura 4.24- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.....	84
Figura 4.25- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.....	85
Figura 4.26- Gráfico do desempenho do Algoritmo Levenberg-Marquardt durante treinamento.	87
Figura 4.27- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o treinamento da Operação Descontínua - com dados com variação na entrada.	87
Figura 4.28- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste do modelo da Operação Descontínua - com dados com variação na entrada.	88
Figura 4.29 - Gráfico do desempenho do Algoritmo Parada Antecipada durante treinamento...	89
Figura 4.30- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Descontínua - com dados com variação na entrada.	90
Figura 4.31- Gráfico do desempenho do Algoritmo Regularização Bayesiana durante treinamento.	92
Figura 4.32 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Descontínua - com dados com variação na entrada.	93
Figura 4.33 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Descontínua - com dados com variação na entrada.....	94
Figura 4.34- Gráfico do desempenho do Algoritmo Levenberg-Marquardt durante treinamento.	96
Figura 4.35 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Descontínua - com dados sem variação na entrada.....	96
Figura 4.36- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Descontínua - com dados sem variação na entrada.....	97
Figura 4.37- Gráfico do desempenho do Algoritmo Parada Antecipada durante treinamento...	98
Figura 4.38- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Descontínua - com dados sem variação na entrada.....	99
Figura 4.39- Gráfico do desempenho do Algoritmo Regularização Bayesiana durante treinamento.	102
Figura 4.40 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Descontínua - com dados sem variação na entrada.....	102
Figura 4.41 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Descontínua - com dados sem variação na entrada.	103
Figura 4.42- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua com Coeficiente de Partição, Vazão Total das Fases e Freqüência.	107
Figura 4.43- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua com Coeficiente de Partição, Vazão Total das Fases e Freqüência. ..	108
Figura 4.44- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua com Freqüência e Vazão Total das fases.....	109
Figura 4.45 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua com Freqüência e Vazão Total das fases.	110
Figura 4.46 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua com Freqüência e Vazão Total das fases.....	111

Figura 4.47- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua com Freqüência e Vazão Total das fases.....	112
Figura 4.48- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua com Coeficiente de Partição e Freqüência.....	113
Figura 4.49- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua com Coeficiente de Partição e Freqüência.	114
Figura 4.50 - Gráfico do desempenho do Algoritmo Levenberg-Marquardt durante treinamento.	116
Figura 4.51 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua - com dados com variação na entrada.....	116
Figura 4.52 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua- com dados com variação na entrada.....	117
Figura 4.53 - Gráfico do desempenho do Algoritmo Parada Antecipada durante treinamento.	118
Figura 4.54 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua - com dados com variação na entrada.....	119
Figura 4.55 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua - com dados com variação na entrada.....	119
Figura 4.56 - Gráfico do desempenho do Algoritmo Regularização Bayesiana durante treinamento.	121
Figura 4.57 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua- com dados com variação na entrada.....	121
Figura 4.58 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua- com dados com variação na entrada.....	122
Figura 4.59 - Gráfico do desempenho do Algoritmo Levenberg-Marquardt durante treinamento.	123
Figura 4.60 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua - com dados sem variação na entrada.....	124
Figura 4.61 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua- com dados sem variação na entrada.....	124
Figura 4.62 - Gráfico do desempenho do Algoritmo Parada Antecipada durante treinamento.	126
Figura 4.63 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua- com dados sem variação na entrada.....	126
Figura 4.64 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua - com dados sem variação na entrada.	127
Figura 4.65 - Gráfico do desempenho do Algoritmo Parada Antecipada durante treinamento.	129
Figura 4.66 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua- com dados sem variação na entrada.....	129
Figura 4.67 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua - com dados sem variação na entrada.	130

ÍNDICE DE TABELAS

Tabela 2.1: Variáveis estudadas por Rabelo (1999):.....	25
Tabela 2.2 - Valores das variáveis estudadas:.....	26
Tabela 2.3 - Principais funções de ativação:.....	33
Tabela 4.1- Variáveis de entrada da Operação Descontínua.....	55
Tabela 4.2 - Variáveis de saída da Operação Descontínua.....	55

Tabela 4.3 - Variáveis de entrada da Operação Contínua.....	55
Tabela 4.4 – Variáveis de saída da Operação Contínua	56
Tabela 4.5 - Treinamento com Algoritmo Levenberg-Marquardt	86
Tabela 4.6 - Treinamento com Algoritmo Parada Antecipada.....	89
Tabela 4.7 - Treinamento com Algoritmo Regularização Bayesiana.....	92
Tabela 4.8 - Treinamento com Algoritmo Levenberg-Marquardt	95
Tabela 4.9 - Treinamento com Algoritmo Parada Antecipada.....	98
Tabela 4.10- Treinamento com Algoritmo Regularização Bayesiana.....	101
Tabela 4.11 - Treinamento com Algoritmo Levenberg-Marquardt.....	115
Tabela 4.12 - Treinamento com Algoritmo Parada Antecipada.....	117
Tabela 4.13 - Treinamento com Algoritmo Regularização Bayesiana.....	120
Tabela 4.14 - Treinamento com Algoritmo Levenberg-Marquardt.....	122
Tabela 4.15 - Treinamento com Algoritmo Parada Antecipada.....	125
Tabela 4.16 - Treinamento com Algoritmo Regularização Bayesiana.....	128

NOMENCLATURA

a - componente do vetor de entrada

A - amplitude de pulsação (cm ou m)

b - saída do neurônio

c - saída calculada

C_b - concentração da proteína na fase fundo (M/L^3)

C_f - fração mássica de soluto na alimentação

C_r - fração mássica de soluto no refinado

C_t - concentração da proteína na fase topo (M/L^3)

d - saída desejada

D - conjunto de dados

e - erro

E - erro quadrático

e - área livre da secção transversal fracional

E^* - Eficiência de Murphree (%)

f - frequência (pulsos/s)

$f(\)$ - função de ativação

g - aceleração da gravidade = $9.81(m/s^2)$

H - matriz de Hessian

i - i-ésimo componente da camada de entrada

I - vetor de entrada

I_R - Índice de recuperação de soluto

j - j-ésimo componente da camada intermediária

$J(\bar{y})$ - matriz Jacobiana dos erros

-
- k - k -ésimo componente da camada de saída
- K - coeficiente de partição da proteína
- k - constantes
- L - número de neurônios camada de entrada
- l - comprimento característico (m)
- M - modelo de rede neural usado
- M - número de neurônios da camada de saída
- n - n -ésimo componente da camada intermediária
- N - número total de parâmetros da rede
- N - número de neurônios da camada intermediária
- $P(D|\alpha, \beta, M)$ - fator de normalização
- $P(D|w, \beta, M)$ - função de probabilidade dos dados
- $P(w|\alpha, M)$ - probabilidade anterior
- $P(w|D, \alpha, \beta, M)$ - probabilidade posterior
- T - transposta
- v - pesos da camada intermediária
- v - velocidade das fases (cm/min ou m/min)
- V - volume (m^3)
- w - pesos da camada de saída
- w - vetor peso da rede
- w^{MP} - ponto mínimo da densidade posterior
- y_f - concentração do soluto na alimentação (fração molar)
- y_r - concentração do soluto no refinado (fração molar)

y_r^* - concentração do soluto na fase refinado em equilíbrio com a fase extrato (fração molar)

LETRAS GREGAS

α - coeficiente do momentum

β - constante de proporcionalidade

δ - declínio do gradiente

∂ - derivada parcial

ϵ - erro da saída

η - taxa de aprendizado

Δ - variação (anterior – atual)

ϕ - fração de retenção da fase dispersa

ρ - massa específica (kg/m^3)

γ - número efetivo de parâmetros

α - parâmetro da função objetivo

β - parâmetro da função objetivo

ξ - tensão interfacial (mN/m)

μ - viscosidade (mPa.s)

ABREVIATURAS

D - dados experimentais

Disp - dispersão dos dados - somatório dos erros do modelo

DP D - Desvio padrão dos dados experimentais

DP R - Desvio padrão dos dados da rede

IA - Inteligência Artificial

NEP - Número efetivo de parâmetros

PEG- Polietileno Glicol

R - Dados das Redes

RNA - Redes Neurais Artificiais

SQE - Soma quadrática dos erros

SQP - Soma quadrática dos pesos

1.0 Introdução

Nos últimos anos, a extração líquido-líquido em sistemas bifásicos aquosos ganhou uma maior atenção dos pesquisadores, projetistas e das indústrias químicas especialmente para a produção de produtos biosintéticos. Em comparação com técnicas clássicas, este tipo de separação é mais adequado por diversas razões como : simplicidade, alta capacidade, alto rendimento e possibilidade de operação em contínuo.

A modelagem da extração líquido-líquido pode ser feita através de abordagens fenomenológicas ou empíricas. Para a modelagem fenomenológica é necessária a compreensão dos fenômenos físicos e químicos, e nem todos são conhecidos. Na abordagem empírica não é necessário o conhecimento dos fenômenos envolvidos. Uma das abordagens empíricas é o uso de redes neurais artificiais.

As redes neurais têm a habilidade de ‘aprender’ comportamentos complexos do sistema físico. O aprendizado é um procedimento pelo qual a rede aproxima uma função mapeando um conjunto de dados de entrada-saída, dado um conjunto de entradas observadas e suas correspondentes saídas. As redes neurais podem ser usadas como ‘caixas pretas’, não é necessário nenhum conhecimento sobre o comportamento do processo. O modelo do processo está baseado somente nas observações dos dados de entrada-saída.

Objetivos

- * Aprendizado da ferramenta Redes Neurais para a modelagem de Sistemas Bifásicos.
- * Modelagem via Redes Neurais da extração do citocromo b5 em operação descontínua.
- * Modelagem via Redes Neurais da extração do citocromo b5 em operação contínua de uma coluna de campânulas pulsantes.

Apresentação do Trabalho

A presente dissertação encontra-se dividida em seis capítulos. O 1^o capítulo descreve uma breve introdução ao problema abordado. O 2^o capítulo aborda a revisão de bibliografia e os fundamentos teóricos. O 3^o capítulo descreve os materiais e métodos utilizados. No 4^o capítulo faz-se a discussão dos resultados e no 5^o capítulo as conclusões obtidas. No 6^o capítulo encontram-se as referências bibliográficas.

2.0 Revisão de Literatura e Fundamentos Teóricos

2.1 Proteínas

As proteínas são macromoléculas cuja estrutura é constituída de polipeptídios, que são cadeias extremamente longas de muitas unidades de aminoácidos, unidas por ligações peptídicas (Lehninger, 1986). As proteínas não são apenas as moléculas mais abundantes, mas extremamente versáteis em suas funções. Atualmente, elas são comercialmente muito importantes, principalmente nas indústrias de alimentos e farmacêutica. Todos os vinte aminoácidos encontrados em proteínas têm em comum um grupo carboxila e um grupo amina, ligados ao mesmo átomo de carbono.

A cadeia polipeptídica de diferentes proteínas pode ter um número qualquer de resíduos de aminoácidos, de um mínimo de aproximadamente 100 até um máximo próximo a 1800. A massa molar das proteínas pode variar de cerca de menos de 12000 Da, para pequenas proteínas, as massas molares maiores que 10^6 Da, no caso de proteínas com cadeias polipeptídicas muito longas ou com várias cadeias polipeptídicas (Lehninger, 1986).

As proteínas podem ter quatro tipos de estruturas. As definições de Murray et al (1994) encontram-se a seguir.

Estrutura Primária: refere-se à seqüência dos aminoácidos na cadeia e a localização das pontes de dissulfeto, se existirem.

Estrutura Secundária: são as relações no espaço entre os aminoácidos componentes da estrutura primária. Estas estruturas podem ser regulares, por exemplo, α -hélice, ou exibir algumas irregularidades, por exemplo, enovelamento.

Estrutura Terciária: para uma proteína, contendo uma única cadeia de polipeptídios refere-se ao arranjo global e relações entre as várias regiões e vários aminoácidos individuais. A estrutura terciária considera as relações espaciais entre os resíduos de aminoácidos que geralmente podem estar distanciados quando considera-se a estrutura primária.

Estrutura Quaternária: as proteínas apresentam a estrutura quaternária quando são constituídas por duas ou mais cadeias polipeptídicas unidas por pontes de hidrogênio e ligações eletrostáticas formadas entre resíduos da superfície das cadeias polipeptídicas.

Pode-se identificar várias classes de proteínas de acordo com as suas funções biológicas (Lehninger, 1986), entre elas estão, por exemplo, as enzimas, que são proteínas com atividade catalítica.

As proteínas também podem ser divididas em duas classes, baseadas em sua forma e certas características físicas, em proteínas globulares e proteínas fibrosas. Nas proteínas globulares, a cadeia, ou cadeias, polipeptídicas estão enroladas, de modo compacto em forma globular ou esférica. Elas são usualmente solúveis em sistemas aquosos e difundem-se facilmente. Proteínas fibrosas são moléculas insolúveis em água, compridas e filamentosas que tem a cadeia polipeptídica estendida ao longo de um eixo.

2.1.1 Citocromo b5

O citocromo b5 faz parte de um complexo enzimático que catalisa a insaturação de ácidos graxos. Trata-se de uma proteína de membrana constituída por uma única cadeia polipeptídica é cuja estrutura terciária apresenta dois domínios distintos. Um deles de natureza globular e hidrofílica, contém um grupo heme que está envolvido nas reações de oxi-redução nas reações de insaturação de ácidos graxos. Pelo contrário, o outro possui natureza hidrofóbica e é responsável pela ligação da proteína à membrana do retículo endoplasmático (Porto, 1998).

O domínio hidrofílico do citocromo b5 de rato foi clonado em *Escherichia Coli* (TB1) através da síntese de novo do gene e da sua introdução no plasmídeo PUC 13 (Von Bodman, 1986), o que permite a sua produção por fermentação, acumulando-se esta no espaço intracelular. A proteína assim obtida possui uma massa molar de 13600 Da e um ponto isoelétrico de 4,4. Ao longo deste trabalho, o domínio hidrofílico do citocromo b5 de rato é designado apenas por citocromo b5 (Porto, 1998).

2.2 Etapas de Separação em Processos Biotecnológicos

Na obtenção de substâncias através de técnicas biotecnológicas é necessário separar misturas de várias proteínas, além de outros contaminantes. Existem várias técnicas com esta finalidade. Entre elas está a diálise, a eletroforese, a filtração em gel e cromatografia, entre outros. Alguns fatores que limitam o uso de algumas destas técnicas são os baixos rendimentos, alto custo e dificuldades de utilização em larga escala. Uma seqüência geral tradicional para a purificação de proteínas intracelulares é mostrada a seguir (Hustedt et al, 1985):

-
- Rompimento Celular
 - Remoção de fragmentos das células:
 - Centrifugação
 - Filtração
 - Concentração:
 - Ultrafiltração
 - Tratamentos Térmicos e ou pH
 - Precipitação por sal, solventes ou polímeros
 - Remoção seletiva de ácidos nucléicos e outras frações
 - Purificação de alta resolução:
 - Cromatografias (por exemplo, cromatografia em gel, de troca iônica, de adsorção, de afinidade e de hidrofobicidade)
 - Concentração Final.

A purificação de proteínas, do ponto de vista técnico, encontra muitas dificuldades e exige um elevado número de etapas. Por exemplo, a remoção de fragmentos da célula é difícil devido ao pequeno tamanho das partículas e à viscosidade da solução. As etapas de precipitação podem levar a baixos rendimentos e reprodutibilidade limitados. O uso de procedimentos cromatográficos, em muitos casos, é limitado pela escala das operações e pelo custo das resinas (Husted et al, 1985). Por isso, a extração líquido-líquido tem despertado o interesse de pesquisadores como mais uma etapa no processo de purificação e separação de proteínas. Deve ser lembrado que a extração líquido-líquido é usada como uma etapa intermediária de separação, que substitui métodos de separação mais caros ou diminui o número de etapas de separação necessárias no processo.

Durante os últimos anos, a purificação e separação de proteínas fundamentadas nos princípios de partição em sistemas de duas fases aquosas têm sido desenvolvidas. Esta técnica de extração parece ser especialmente adequada para as primeiras etapas do procedimento de

separação, mas pode substituir etapas cromatográficas ou ser aplicadas antes da cromatografia (Husted et al, 1985).

2.2.1 Extração Líquido-Líquido

Uma situação muito comum na Engenharia Química é a separação dos constituintes de uma mistura líquida homogênea composta de dois ou mais componentes. Para realizar esta separação existem vários métodos cuja aplicação é limitada pelas características físicas e químicas dos componentes da mistura a ser separada, pelos custos do processo de separação e pelas condições disponíveis para a implantação do processo escolhido.

A extração líquido-líquido, um dos principais métodos de separação adotados industrialmente, é um processo que envolve a transferência de massa entre dois líquidos imiscíveis ou parcialmente miscíveis. Na extração líquido-líquido, a separação de um componente homogêneo ocorre pela adição de um constituinte líquido, insolúvel ou parcialmente solúvel na solução, o solvente, no qual o componente a ser extraído da solução, o soluto, é preferencialmente solúvel. O soluto difunde-se no solvente com uma velocidade característica até atingir concentrações de equilíbrio em cada uma das fases formadas. Este processo de separação é baseado na distribuição do soluto entre duas fases e na miscibilidade parcial dos líquidos.

O conhecimento das relações de equilíbrio de fases é fundamental para análises quantitativas de processos de extração. As relações de equilíbrio de fases são baseadas em princípios termodinâmicos e representadas por diagramas estabelecidos por leis de distribuição dos constituintes entre as fases. Na extração líquido-líquido, o efeito de pequenas variações de pressão sobre o equilíbrio é insignificante, restringindo a atenção à influência da temperatura e da concentração (Coimbra, 1995).

Geralmente, a extração líquido-líquido é empregada nos casos em que os componentes sejam sensíveis à temperatura requerida para a separação, quando o componente desejado seja pouco volátil e esteja presente em pequena quantidade na solução ou quando a separação por outros métodos tenha alto custo, entre outros casos.

Quanto à viabilidade econômica dos processos que empregam a extração líquido-líquido, esta é fortemente influenciada pelos custos do equipamento e pelos custos de

recuperação e reposição do solvente. O custo de operação de um extrator líquido-líquido (consumo de energia mais manutenção) é geralmente negligenciável comparados aos outros fatores (Hanson, 1968 a e b).

Embora aplicada com sucesso há muito tempo, como uma técnica de separação laboratorial, sua utilização a nível industrial iniciou-se nas primeiras décadas do século XX, quando se procurava um método que resolvesse o problema da remoção de hidrocarbonetos aromáticos do querosene. Principalmente depois da Segunda Guerra Mundial, teve grande desenvolvimento na indústria, alcançando os mais diversos setores. Atualmente, é empregada com sucesso, em larga escala, em setores industriais tais como, extração de metais, extração de compostos orgânicos, como ácidos e álcool, purificação de antibióticos, indústria farmacêutica e para tratamentos de efluentes industriais, entre outros. Sua maior aplicação encontra-se na indústria petroquímica. Mais recentemente, a extração líquido-líquido vem sendo estudada e empregada na área de biotecnologia para a separação de biomateriais, como enzimas e proteínas.

A extração líquido-líquido pode ser realizada de diversas formas por meio de vários tipos de equipamentos, mas sempre seguindo os mesmos princípios básicos de contato entre os líquidos, transferência de massa, equilíbrio de fases e separação de fases. Esta separação das fases comumente ocorre pelo processo de decantação (Rabelo, 1995).

2.3 Sistemas de Duas Fases Aquosas

Na escolha de meios de extração para aplicações em biotecnologia, vários critérios devem ser considerados, já que nesta área parâmetros como, solubilidade e estabilidade dos compostos são importantes e não podem ser desprezados. Entre estes critérios pode-se citar (Porto, 1998):

- meio não deve ser tóxico ao sistema biológico nem ao homem;
- a recuperação do bioproduto a partir do meio extrator deve ser fácil;
- deve ter baixo custo e estar disponível comercialmente em grande quantidade;
- ser passível de esterilizar;
- ser imiscível ou parcialmente miscível com soluções aquosas;

- não deve ter a tendência de formar emulsões estáveis com materiais biológicos;
- não ser inflamável.

Além destes fatores, em processos de extração líquido-líquido aplicados a quaisquer sistemas, é imprescindível que os solventes escolhidos formem duas fases (sejam imiscíveis ou parcialmente miscíveis) e tenham densidades diferentes (Husted et al 1985). É importante que a separação das fases seja rápida.

Nos processos biotecnológicos, em que se opera com biomoléculas ou células, existe um número muito limitado de solventes adequados a serem usados. O uso de solventes orgânicos é, normalmente, limitado pelas características hidrofílicas dos produtos de fermentação. Isso leva à necessidade de utilização de elevadas razões entre as fases orgânica e aquosa, devido aos baixos coeficientes de partição dos produtos em relação ao solvente orgânico. Além disso, os solventes orgânicos são geralmente tóxicos para as proteínas e também provocam a desnaturação das mesmas. Assim a introdução dos sistemas de duas fases aquosas é uma alternativa que possibilita o emprego de extração líquido-líquido nestes processos, já que estes sistemas ajustam-se aos critérios requeridos nas bioseparações.

Os sistemas de duas fases aquosas formam-se pela adição de soluções aquosas de dois polímeros hidrofílicos, como por exemplo, polietileno glicol (PEG) e dextrana ou de um polímero e um sal, como por exemplo, o PEG e fosfatos de potássio, em concentrações acima de seus valores críticos, quando separam-se em duas fases, espontaneamente. Nos exemplos citados, a fase mais leve, que tem a menor densidade, é rica em polietileno glicol, enquanto a fase mais pesada, que tem a maior densidade, é rica em dextrana ou sais. Os polímeros e os sais são solúveis em água, mas são incompatíveis entre si e se separam em duas fases (Albertsson, 1986). Eles constituem um meio suave e protetor, adequado para a extração de substâncias de origem biológica, pois as fases são compostas de cerca de 70% a 90% de água, o que proporciona um ambiente ameno e adequado para o trabalho com compostos biologicamente ativos, preservando a sua estabilidade molecular e permitindo o seu processamento neste meio (Coimbra, 1995).

Albertsson constatou que sistemas formados por polímeros solúveis e solventes orgânicos, assim como sistemas formados por polímeros hidrofílicos e água, também possibilitavam a partição de materiais biológicos, ou seja, permitiam uma terceira substância

introduzida ao sistema fosse coletada, preferencialmente, em uma das fases, de acordo com ajustes de parâmetros físico-químicos. Devido a esta característica, os sistemas de duas fases aquosas estão sendo empregados no isolamento e purificação de biomoléculas de importância comercial, tais como, células, proteínas, enzimas, vírus, fragmentos de membranas e organelas celulares. É possível obter-se separações bastante seletivas de substâncias, usando-se sistemas aquosos de polímeros, sob condições não desnaturantes.

Os sistemas de duas fases aquosas formados por PEG-Dextrana-Água e PEG-Sal - Água têm sido, nos últimos anos, os mais freqüentemente estudados e utilizados para purificação de um grande número de biomoléculas (Diamond et al, 1992).

Geralmente, considera-se que a separação de moléculas em sistemas de duas fases aquosas é dependente das características da superfície molecular dos compostos a serem particionados, tais como: carga, tamanho e propriedades hidrofóbicas (Albertsson 1971; LaMarca et al, 1990).

O efeito da massa molar do polímero, concentração, densidade das fases e a presença de sais em sistemas de duas fases aquosas têm sido estudados por vários autores (Albertsson, 1971, 1986; Johanson, 1985; Bamberger et al, 1984; Zaslavsky et al, 1992; Carlson, 1998; Huddleston et al, 1991; Asenjo et al, 1990; Sarmiento et al, 1994).

A extração com sistemas de duas fases aquosas oferece vantagens para o processamento em larga escala. Algumas delas são: elevado rendimento, faixa de trabalho próxima ao equilíbrio, fácil ampliação de escala e a possibilidade de processamento contínuo (Kim et al, 1998). Esses fatores justificam o interesse na aplicação de sistemas de duas fases aquosas. A maioria dos estudos concentra-se na análise dos fundamentos da separação de fases e da partição de proteínas, na redução dos custos do processamento, no aumento da seletividade da extração (por exemplo, pela adição de ligantes), na pesquisa de novos componentes formadores das fases, especialmente para substituir a dextrana, que é um polímero de custo elevado e no desenvolvimento de novos sistemas (Szlak e Giuliano, 1998), na operação em múltiplos estágios (Coimbra 1995), na ampliação de laboratório para planta piloto ou larga escala (Husted et al, 1985; Papamichael et al, 1992) e na modelagem do processo de partição dos componentes (Grossman et al, 1993; Eiteman, 1994).

2.3.1 Partição de Proteínas em Sistemas de Duas Fases Aquosas

Os fundamentos de partição de biomoléculas entre duas fases ainda não são totalmente compreendidos, apesar do progresso alcançado com a modelagem termodinâmica da distribuição de proteínas na interface dos sistemas de duas fases aquosas. Existem duas diferentes aproximações para a abordagem termodinâmica, a primeira utiliza o tratamento rigoroso para a modelagem do equilíbrio (Baskir et al, 1989) e a outra, métodos de contribuição de grupos (Kang e Sandler, 1988).

A tendência de separação de fases apresentada por dois polímeros quando adicionados em um solvente comum, ocorre porque a baixa concentração molar dos polímeros na solução (tipicamente menor que 0,05 mol/L) leva a um pequeno ganho de entropia durante a mistura. Por outro lado, cadeias poliméricas têm uma área superficial por molécula maior do que de moléculas de baixa massa molar, tanto que as energias de interação entre os dois polímeros se sobrepõem à energia de Gibbs do sistema. Estes fatores levam à formação de duas fases em sistemas ternários polímero-polímero-água, em baixas concentrações dos polímeros.

Há vários fatores que influenciam a partição de biomoléculas em sistemas de duas fases aquosas. Estes fatores são inerentes ao próprio sistema (tipo de componentes do sistema, massa molar do sistema, massa molar do polímero, concentração do polímero e dos sais, tipo de íon presente, força iônica e pH) e ao biocomposto que é distribuído (massa molar, carga, hidrofobicidade e conformação, presença de ligantes específicos, entre outros). A seleção de propriedades dos sistemas de duas fases aquosas apropriados para a purificação de uma proteína específica é ainda muito empírica, embora existam regras gerais de como a proteína sofre a partição, com relação ao efeito das características do polímero e da composição iônica da proteína (Cascone et al, 1991).

A modelagem quantitativa da partição de proteínas em sistemas de duas fases aquosas representa um complexo problema porque o comportamento do sistema depende de muitos fatores, que já foram citados anteriormente e das interações entre eles: interações dos polímeros com as cadeias das proteínas e com diferentes sais, interações polímero-polímero, sal- proteína, proteína-proteína e, quando usados, as interações dos ligantes com os outros componentes presentes. Estudos empíricos com sistemas de duas fases aquosas mostraram ser a distribuição da proteína função desses fatores.

A existência de tantas variáveis torna extremamente difícil a previsão teórica do coeficiente de partição de um dado soluto, conduzindo muitas vezes, ao trabalho experimental (Sebastião et al, 1994; Porto, 1998).

Considerando que C_t e C_b sejam as concentrações da proteína nas fases topo e fundo, o coeficiente de distribuição das proteínas é definido por:

$$K = \frac{C_t}{C_b} \quad (2.1)$$

Modelos quantitativos para a predição do comportamento do sistema em função do tipo de polímero, de biomolécula, da composição das fases, do pH e da temperatura não estão disponíveis. Além disso, os mecanismos fundamentais que governam o comportamento dessas variáveis ainda não foram satisfatoriamente entendidos (Baskir et al, 1989). Uma vez mais ressalta-se a importância do desenvolvimento da termodinâmica da extração em sistemas de duas fases aquosas (Kula, 1990).

O coeficiente de partição de uma proteína em um sistema de duas fases aquosas, tendo o PEG como fase superior, geralmente, aumenta com a diminuição da massa molar do PEG. Este fato é tanto mais pronunciado quanto maior for a massa da biomolécula a sofrer a partição (Albertsson et al, 1987).

A adição de sais, mesmo em concentrações milimolares, influencia fortemente a partição dos materiais carregados eletricamente. A alteração do pH do sistema contendo sal poderá alterar a partição pela mudança na carga da proteína (Johansson, 1985). Como a carga, na maioria dos biomateriais, é dependente do pH, a escolha deste e de um sal pode constituir um modo efetivo de ajuste da partição (Porto, 1998).

O fato da partição depender de um grande número de fatores distintos confere considerável versatilidade aos sistemas de duas fases na separação de misturas de componentes.

A seletividade na partição de proteínas tem ainda que ser mais estudada e entendida. Um melhor conhecimento do comportamento de proteínas em sistemas de duas fases aquosas também levará à habilidade de prever a partição de proteínas-alvo, frequentemente encontradas em uma mistura de proteínas heterogênea e complexa (Franco, 1992).

A partição em sistemas de duas fases aquosas é principalmente, um processo onde os grupos expostos das moléculas entram em contato com os componentes das fases e é, portanto, dependente da superfície (Albertsson, 1986).

2.4 Equipamentos Usados nos Processos de Extração

A escolha de um extrator é função da natureza dos produtos e das características físicas e químicas do sistema em estudo, além de ser função do desempenho do equipamento, considerando-se as condições de extração.

Para o desenvolvimento de equipamentos de extração líquido-líquido, deve-se avaliar além do sistema de trabalho, os dados de equilíbrio e de transferência de massa nas temperaturas de interesse.

A utilização de extratores verticais nos processos de extração líquido-líquido é justificada pelo fato desses equipamentos ocuparem pequena área, terem boa eficiência de separação e apresentarem facilidades de operação e manutenção. Além disso, nesses equipamentos, o processo de transferência de massa e separação das fases são mais rápidos e o sistema pode atingir o equilíbrio ou aproximar-se dele em um período de tempo menor do que outros tipos de equipamentos.

Existe um grande número de equipamentos desenvolvidos para a aplicação da extração líquido-líquido, que podem ser classificados de acordo com os princípios gerais de operação e construção. Os extratores dividem-se em duas categorias, de acordo com o tipo de contato entre as fases: sistemas de contato e separação de fases em estágios e sistemas de contato diferencial.

2.4.1 Coluna Agitada por Campânulas Pulsantes

Entre as colunas de contato diferencial foi desenvolvida uma micro-coluna de campânulas pulsantes, para obter-se um equipamento para a realização de processos de extração líquido-líquido, visando a separação e purificação de biomoléculas.

O uso de campânulas em colunas pulsadas foi proposto visando-se obter uma agitação eficiente, mas suave, para a aplicação em processos de separação de compostos lábeis, como as proteínas. Sendo a agitação suave, pretende-se evitar a desnaturação de biomoléculas, que ocorre quando são introduzidas ao sistema excessiva energia mecânica e agitação extremamente vigorosa.

Além disso, as campânulas aumentam o tempo de contato entre as fases na coluna e fazem com que ocorra uma dispersão uniforme das gotas, o que favorece a transferência de massa, fazendo com que o sistema se aproxime do equilíbrio (Rabelo, 1999).

Com o sistema de agitação composto por campânulas pulsantes, as fases que entram na coluna comportam-se da maneira mostrada na figura 2.1. Como pode ser visto, a fase leve entra na micro-coluna pelo fundo e fica retida na campânula. A peneira, da qual a campânula é constituída, não deixa a maior parte da fase leve passar espontaneamente. A campânula retém a fase leve por um tempo maior do que outros dispositivos comumente usados para promover a agitação. Deste modo, ocorre um aumento do tempo de contato entre as fases (Rabelo, 1999).

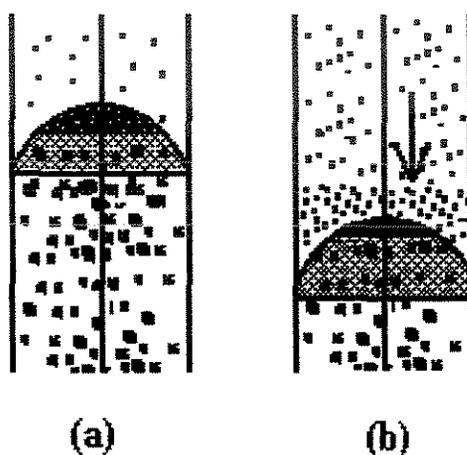


Figura 2.1 – Agitação através de campânulas pulsantes numa coluna de extração líquido-líquido pulsada. (a) campânula parada durante o intervalo entre pulsos. (b) movimento descendente durante o pulso.

Durante o pulso, quando a campânula desce, o líquido mais leve (de menor densidade) que fica concentrado no interior da campânula, é forçado a passar pelos furos da peneira. A fase leve é “quebrada” em gotas bem pequenas. Uma importante característica da agitação obtida através do uso de campânulas pulsantes é a formação de uma dispersão, bastante uniforme e de gotas pequenas, da fase leve na fase contínua, como pode ser visto na figura 2.1. Essas características são favoráveis à transferência de massa. Quando a campânula sobe, o movimento ascendente faz com que as fases misturem-se ainda mais (Rabelo, 1999).

As campânulas devem ser construídas de material inerte que não reaja com as soluções usadas na extração (Rabelo, 1999).

2.4.2 Descrição do Equipamento

Um esquema da micro-coluna de campânulas pulsantes é mostrado na figura 2.2. Esta micro-coluna é constituída de um tubo de vidro de 19 cm de altura; 2,84 cm de diâmetro externo e 2,54 cm de diâmetro interno.

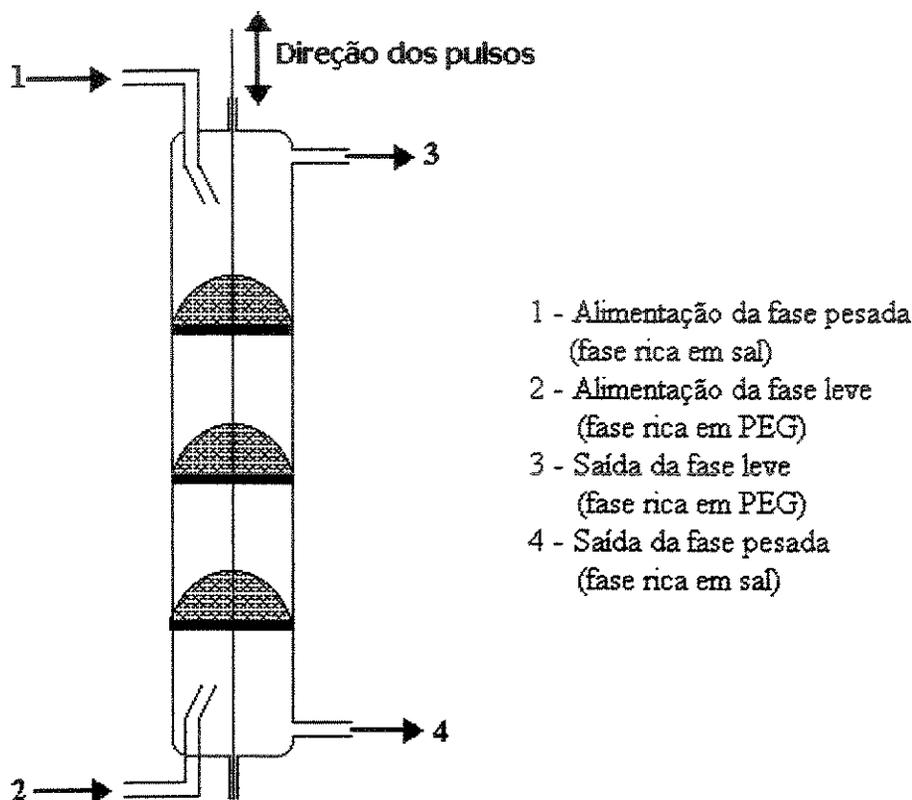


Figura 2.2 – Esquema da micro coluna de campânulas pulsantes.

No centro desta micro-coluna, está localizada uma haste de aço inox, na qual estão soldadas três campânulas, distanciadas 4 cm uma da outra. O diâmetro da base da campânula é 2,49 cm. A peneira de aço inox, da qual a campânula é construída, tem MESH 24 e, portanto, possui uma área livre para o escoamento de 38%. Durante um pulso, a campânula movimentar-se aproximadamente 2,8 cm para baixo e 2,8 cm para cima, o que faz com que a amplitude do pulso seja de aproximadamente 5,6 cm.

Os bocais de alimentação e saída das fases são de aço inox e possuem um diâmetro interno de 3,65 mm e 6,85 mm de diâmetro externo. Os pulsos foram fornecidos por um dispositivo que promove os movimentos alternativos (movimenta o eixo para cima e para baixo) e controlados por um controlador da frequência de pulsação. A entrada das duas fases (fase leve e fase pesada) é feita por meio de uma bomba peristáltica de quatro pontos, previamente calibrada.

Os tubos para conduzir as entradas e saídas das fases são de viton, que é um material resistente à ação da maioria dos solventes. O viton foi escolhido para permitir que outros sistemas sejam usados, já que os sistemas de duas fases aquosas não são corrosivos.

A temperatura dos experimentos foi mantida em $(24 \pm 3)^{\circ}\text{C}$.

A extração do citocromo b5 nesta coluna foi estudada por Rabelo (1999) onde foi analisada a influência de algumas variáveis operacionais sobre a hidrodinâmica e transferência de massa, possibilitando a identificação das melhores condições de operação da coluna. As variáveis estudadas foram:

Tabela 2.1: Variáveis estudadas por Rabelo (1999):

Variáveis Estudadas	Valores Assumidos
Vazão Total de Alimentação (mL/min)	4,7; 6,5; 10,9
Frequência de pulsação (pulsos/s)	0,1; 0,143; 0,2; 0,33; 1,0
Razão entre as Vazões das Fases (V_d/V_c)	Aproximadamente 1

Neste estudo foram avaliados a fração de retenção da fase dispersa, Índice de Recuperação de Sóluto e Eficiência de Separação.

2.5 Extração do Citocromo b5 em Operação Descontínua

Rabelo (1999) estudou a extração descontínua do citocromo b5 a fim de encontrar as condições relativas ao sistema de duas fases aquosas, que levassem à obtenção dos melhores resultados de extração. As condições analisadas foram comprimento da linha de amarração (concentração de PEG e sais – fosfatos de potássio) e massa molar do polímero usado (PEG). A melhor condição foi aplicada na extração em contínuo na coluna de campânulas pulsantes.

Os valores das variáveis estudadas foram:

Tabela 2.2 - Valores das variáveis estudadas:

Massa Molar do PEG	PEG / Fosfatos de Potássio (%p/p)			
	Linhas de Amarração			
	0	1	2	3
400	*	16,7/14,8	17,7/15,7	19,7/17,7
1000	*	16,2/14,3	17,7/15,7	19,7/17,7
1500	*	16,2/14,3	17,7/15,7	19,7/17,7
3350	*	14,0/11,8	17,7/15,7	19,7/17,7
8000	11,8/9,8	14,0/11,8	•	•

* Sistemas homogêneos – formação de uma única fase.

• Não foi possível a dissolução dos componentes das fases.

Na maioria dos casos, para cada tipo de sistema, as concentrações do polímero e do sal foram escolhidas de forma a obter três linhas de amarração: uma próxima ao ponto crítico, outra intermediária e uma mais afastada do ponto crítico.

2.6 Eficiência de Extração

Dois aspectos distintos sobre a eficiência de extração devem ser considerados: o primeiro, diferencia o comportamento real e ideal e o segundo trata a eficiência em termos de mecanismos de transporte de massa. Baseados nestes aspectos, várias definições de eficiência têm sido propostas (Lo et al, 1982).

A eficiência definida por Kawase (1990) é baseada nas concentrações de soluto em uma mesma fase, na entrada e na saída da coluna. Ela é equacionada da seguinte maneira:

$$I_R = \frac{C_f - C_r}{C_f} \quad (2.2)$$

onde: C_f é a fração mássica de soluto na alimentação e

C_r é a fração mássica de soluto no refinado.

Esta definição matemática pode ser chamada de Índice de Recuperação do Solute, percentagem de extração ou percentagem de massa extraída.

A segunda definição de eficiência que pode ser usada, é uma correlação experimental que descreve a eficiência de separação do extrator em estudo, equacionada da seguinte forma:

$$E^* = \frac{y_f - y_r}{y_f - y_r^*} \quad (2.3)$$

Murphree (1925) apresentou esta definição matemática de eficiência, chamada Eficiência de Murphree.

Esta equação representa a relação entre a eficiência obtida no equipamento (eficiência real) e a que seria obtida se o tempo de contato entre as fases fosse suficiente para atingir o equilíbrio (eficiência ideal), portanto representa o desvio em relação ao equilíbrio. Esta equação é muito importante para fins de comparação e projeto de equipamentos de extração. Nesta definição, y_r^* é a concentração do soluto (fração molar) que se encontraria na fase refinado se o sistema estivesse em equilíbrio com a fase extrato.

2.7 Fração de Retenção da Fase Dispersa

A transferência de massa entre as fases líquidas em escoamento em uma coluna de extração depende, entre outros fatores, da área interfacial de contato entre as fases contínua e dispersa. A área interfacial está relacionada com o tamanho das gotas e com a retenção da fase dispersa no interior da coluna. A retenção da fase dispersa é expressa pela fração volumétrica da fase dispersa contida no volume total da coluna, sendo de grande importância no projeto de colunas de extração líquido-líquido por influenciar a transferência de massa entre as fases.

A fração de retenção da fase dispersa, que é a fração volumétrica da fase dispersa no interior do equipamento, é dada pela seguinte expressão:

$$\phi = (\text{Volume da Fase Dispersa}) / (\text{Volume Total da Coluna}) \quad (2.4)$$

Kumar e Hartland (1988) usaram resultados experimentais publicados para a fração de retenção da fase dispersa em colunas de extração de pratos perfurados pulsados no desenvolvimento de uma correlação para prever a fração de retenção da fase dispersa, nas regiões de operação de mistura-decantação, transição e emulsão, a partir de propriedades físicas e variáveis de operação. Adicionalmente, foi proposta uma correlação que prediz os limites entre as regiões de operação de mistura-decantação e de transição.

A equação 2.5 representa a correlação proposta pelos autores para a predição da fração de retenção da fase dispersa, em colunas de pratos perfurados pulsados. Esta equação considera as condições de operação da coluna e as propriedades físicas do sistema e foi originada nos

dados publicados com e sem transferência de massa para 23 sistemas líquido-líquido a partir de 15 diferentes fontes de dados.

$$\phi = k_1 \cdot \exp[k_2 |A.f - (A.f)_t|] v_d^{0,86} \cdot (v_c + v_d)^{0,28} \cdot \Delta\rho^{-0,30} \cdot \rho_d^{-0,93} \cdot \mu_d^{0,77} \cdot \alpha^{-0,56} \cdot h^{-0,56} \quad (2.5)$$

É assumido que o valor do produto amplitude-freqüência, $(A.f)_t$, pode ser descrito pela relação:

$$(A.f)_t = 9,69 \times 10^{-3} \left(\frac{\xi \cdot \Delta\rho^{1/4} \cdot e}{\mu_d^{3/4}} \right)^{0,33} \quad (2.6)$$

$(A.f)_t$ marca a transição da região de mistura-decantação para a região de transição.

Os valores das constantes k_1 e k_2 para os três casos: sem transferência de massa, transferência de massa da fase contínua para a fase dispersa (c→d) e na direção oposta (d→c) são dadas a seguir:

	k_1	k_2
sem transferência de massa	$2,10 \times 10^6$	44,53
transferência (c→d)	$2,14 \times 10^6$	44,53
transferência (d→c)	$1,10 \times 10^6$	50,56

A fração de retenção da fase dispersa, ϕ , é considerada como sendo função de ε (dissipação de energia por unidade de massa), das velocidades da fase contínua e dispersa (v_c e v_d), das propriedades físicas ($\rho_c, \Delta\rho, \mu_c, \mu_d$ e γ), da área livre da secção transversal fracional (e), do comprimento característico (l) e da constante da gravidade (g). O comprimento característico l , é considerado como a altura de um compartimento para colunas agitadas em estágios.

2.8 Redes Neurais Artificiais

Uma Rede Neural Artificial pode ser considerada como sendo um sistema dinâmico não linear de grande escala, consistindo de um grande número de elementos simples de processamento, chamado nós ou neurônios, os quais estão interconectados de alguma maneira através de pesos ajustáveis. As principais características das redes neurais são o processamento paralelo, memória associativa e distributiva, aprendizado e auto-organização, características que são semelhantes às redes neurais biológicas.

2.8.1 O Modelo Computacional das RNA's

Uma rede Neural Artificial (RNA) é uma ferramenta computacional que resultou de pesquisas na área de Inteligência Artificial (IA) que procurava entender e modelar o comportamento do cérebro humano. Com o propósito de desenvolver o mesmo tipo de estrutura para um modelo computacional de comportamento inteligente, neurologistas e pesquisadores de IA propuseram uma rede altamente interconectada de neurônios ou nódulos.

Informações de entrada são colocadas em uma rede de nódulos que interagem matematicamente entre si. Baseado nestas informações, surge um mapeamento do modelo entrada/saída macroscópico esperado, ou seja, as interações entre os nódulos são bem definidas e ajustadas até que as relações entrada/saída sejam obtidas. A interconexão de nódulos forma a RNA, conforme mostra a figura 2.3:

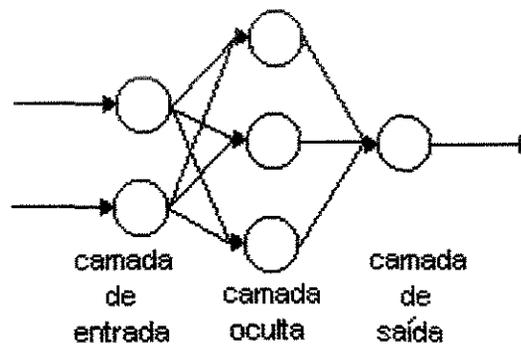


Figura 2.3 - RNA e suas interconexões.

Todas as RNA's devem ter uma camada de neurônios de entrada, uma de saída e, no mínimo, uma camada de neurônios escondida.

Uma RNA pode ser vista como uma caixa preta que recebe informações de entrada específicas em cada nódulo da camada de entrada e então processa estas informações através de suas interconexões entre os nódulos de forma totalmente oculta. Finalmente, a RNA resulta informação final de saída nos nódulos da camada de saída. Resumindo, a camada de entrada recebe informações de fontes externas e passa estas informações para dentro da RNA para processamento. As camadas escondidas recebem informações da camada de entrada e realizam o processamento destas informações. A camada de saída recebe as informações processadas da RNA e envia os resultados para o receptor externo.

Quando a camada de entrada recebe informações da fonte externa, se torna ativada e emite sinais para sua vizinhança. Esta recebe excitação da camada de entrada e mais uma vez emite uma saída para suas vizinhanças. O resultado é um modelo de ativação que se manifesta na camada de saída. Dependendo da força das conexões, sinais podem excitar ou inibir os nódulos.

Uma característica importante das RNA's é que, dentro da rede, o processamento é numérico embora os resultados possam se manifestar simbolicamente.

A rede retém as informações através da magnitude dos sinais que passam através das conexões dos neurônios com sua vizinhança.

Sendo numéricas as informações retidas na RNA, esta pode ser usada como uma ferramenta importante de modelagem empírica multivariável e com algumas vantagens, como sua habilidade de adaptação.

Imitando o processo humano de aprendizagem, as RNA's "aprendem" e "resolvem" problemas por tentativa-e-erro. As RNA's utilizam-se de algoritmos particulares de treinamento ou aprendizagem, onde iterativamente são ajustados os pesos das conexões entre os nódulos até que os pares desejados de entrada/saída sejam obtidos e as relações de causa-efeito possam ser estabelecidas. Se as condições mudarem de tal modo que o desempenho do modelo não seja mais adequado, pode-se submeter a RNA a mais um treinamento sob estas novas condições de entrada/saída para corrigir o desempenho. Assim sendo, pode-se submeter a RNA a uma atualização periódica, resultando num modelo auto-ajustável em linha.

2.8.2 O Neurônio Artificial

O neurônio é uma unidade fundamental de processamento para a operação da rede neural. Na figura 2.4 está representado o modelo, onde podemos identificar três elementos básicos do modelo.

1) Um conjunto de sinapses ou conexões, cada qual está caracterizada por um peso próprio. O sinal a_i da entrada i é conectado ao neurônio j , que é multiplicado pelo peso w_{ij} (Haykin, 1998). Certamente, os valores dos componentes do vetor de entrada A têm efeito sobre a saída b do neurônio, mas alguns componentes adicionais do nódulo também afetam b . um destes componentes é o fator ponderal, w_{ij} , correspondente ao vetor a_i de entrada do j -enésimo

módulo. Estes fatores ponderais, ou pesos, podem assumir efeitos inibitórios ou excitatórios, representando a função dos íons Na^+ e K^+ na Rede Neural Biológica. Se w_{ij} é ajustado de tal modo que o produto $a_i \cdot w_{ij}$ seja positivo (e alto), a tendência é de excitação do neurônio j . Se $a_i \cdot w_{ij}$ for negativo, esta entrada ponderada irá inibir o nódulo. Se $a_i \cdot w_{ij}$ assumir um valor muito pequeno em relação aos outros sinais (ou entrada ponderadas $a_i \cdot w_{ij}$), o efeito será muito pequeno ou nulo sobre o nódulo.

2) Um somatório para somar os sinais de entrada, e multiplicar pelos pesos do neurônio, esta operação constitui uma combinação linear (Haykin, 1998).

3) Uma função de ativação para limitar a amplitude da saída do neurônio. A função de ativação é uma função de ajuste dos limites de saída para valores finitos permitidos (Haykin, 1998).

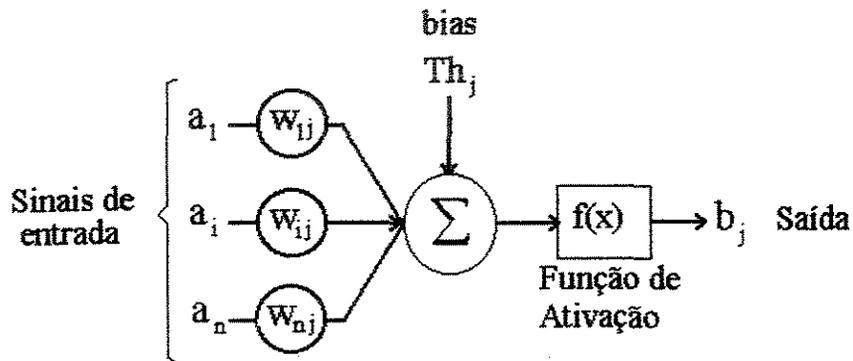


Figura 2.4 - Modelo de um Neurônio Artificial.

O modelo também inclui uma atividade residual (“bias”) aplicada, o bias tem o efeito de diminuir ou aumentar a ativação dependendo se for positivo ou negativo. Se Th_j for grande e positivo, o nódulo tem uma alta ativação residual interna, o que inibe a excitação do mesmo. Ao contrário se Th_j for nulo ou assumir valores negativos, o neurônio tem uma baixa ativação residual interna, sofrendo excitação mais facilmente. Se nenhuma ativação interna for especificada, deve-se assumir Th_j nulo.

Verifica-se então que o neurônio artificial realiza seus cálculos baseado nas informações de entrada. Ele faz o somatório do produto entre os vetores A e W_j , subtrai a ativação residual interna e então passa este resultado para uma forma funcional, $f(\)$, ou seja:

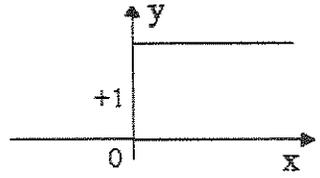
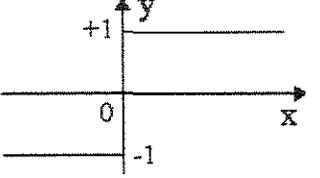
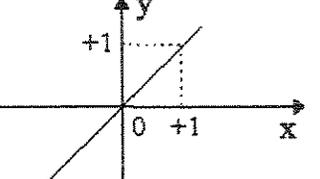
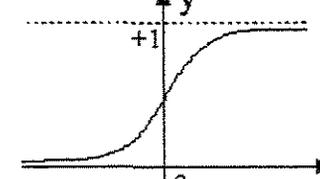
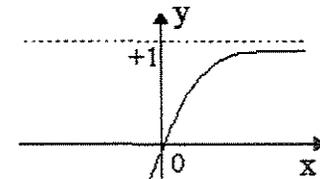
$$f(W_j \cdot A - T_{bj}) = f\left(\sum_{i=1}^n (w_{ij} \cdot a_i) - T_{bj}\right) \quad (2.7)$$

2.8.3 Funções de Ativação

A função de ativação, define a saída do neurônio em termos de saída local. As funções de ativação mais encontradas na literatura são (Tabela 2.3):

- Função degrau, limita a saída do neurônio a apenas dois valores (0 ou 1). Normalmente é utilizada para criar neurônios que tomem decisões binárias, como nos classificadores. É limitada (*"bounded"*), porém não é derivável;
- Função linear, não é limitada. Neurônios com esta função de propagação podem ser utilizados como aproximadores lineares;
- Função sigmoidal, permite que a entrada assuma qualquer valor no intervalo $(-\infty, \infty)$ e os comprime para o intervalo $[0, +1]$. É a função geralmente adotada em redes neurais, em virtude de ser contínua, monotônica, não linear e facilmente diferenciável em qualquer ponto;
- Função tangente hiperbólica, mapeia a entrada dos neurônios no intervalo $[-1, +1]$. Possui as mesmas características e emprego da função logística sigmoidal, possibilitando que as saídas sejam simétricas.

Tabela 2.3 - Principais funções de ativação:

Função	Equação	Gráfico
Degrau	$y = \begin{cases} 1, x > -b \\ 0, x < -b \end{cases}$	
Degrau Simétrico	$y = \begin{cases} 1, x > -b \\ -1, x < -b \end{cases}$	
Linear	$y = x + b$	
Sigmoidal	$y = \frac{1}{1 + e^{-(x+b)}}$	
Tangente Sigmoidal	$y = \frac{e^{(x+b)} - e^{-(x+b)}}{e^{(x+b)} + e^{-(x+b)}}$	

O ideal na função de ativação é que ela seja não linear, limitada, monotônica e continuamente derivável em todos os seus pontos.

2.8.4 Topologia de uma Rede Neural Artificial

A topologia de uma RNA se refere à maneira como os neurônios artificiais estão interconectados e organizados em camadas. Existem três opções de conexões de neurônios, conforme a figura 2.5:

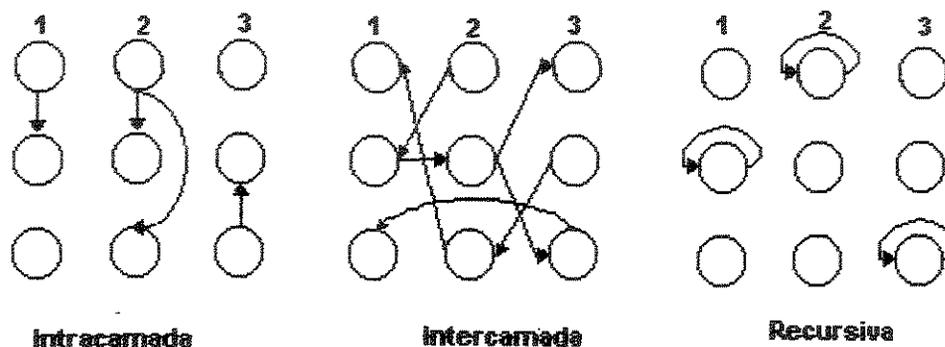


Figura 2.5 - Tipos de conexões entre os neurônios:

Em conexões intracamadas, as saídas de um neurônio alimentam outros neurônios de uma mesma camada. Em conexões intercamadas, as saídas de um neurônio em uma camada alimentam nódulos de outras camadas. E em conexões recursivas, a saída de um neurônio alimenta ele próprio.

Segundo Quantrille e Liu (1991), a conexão intercamadas é particularmente importante para aplicações em Engenharia. Dentro deste esquema de interconexão, existem duas opções: as conexões por retroalimentação e por alimentação direta, conforme a figura 2.6:

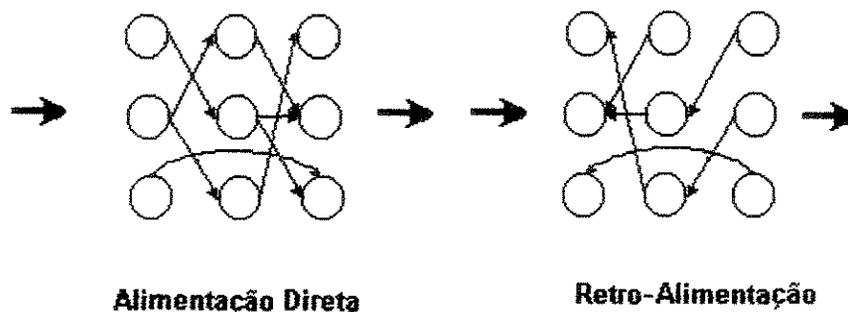


Figura 2.6 - Conexões por Alimentação Direta e Retro-Alimentação.

A topologia ideal deve ser escolhida de acordo com a natureza do problema a ser resolvido. Para modelagens dinâmicas de equipamentos, o que se quer é mapear uma resposta baseada em informações de entrada e para tanto deve ser utilizada a conexão intercambiada com alimentação direta. Além disto, uma rede Alimentação Direta (“Feedforward”) com apenas uma camada oculta é suficiente para a representação de funções contínuas (Cybenko, 1989).

2.9 Treinamento da RNA

A aprendizagem, ou treinamento, é a etapa onde se codificam as informações. Um sistema aprende uma regra se ele codifica esta regra em sua estrutura. Portanto, esta estrutura deve mudar durante o aprendizado desta informação.

Para treinar RNA's, deve-se ajustar os fatores ponderais, W , até que a resposta calculada (informação de saída da rede), baseada em informações de entrada, reproduza a informação de saída conhecida. Se o processo de ajuste dos pesos é feita através de tentativa-e-erro, esta etapa na verdade se denomina aprendizagem (“learning”). Por outro lado, se o ajuste é feito de forma sistemática, esta etapa é denominada treinamento e caracteriza-se então como um problema de otimização onde se busca reduzir erros de predição.

O treinamento precisa produzir uma RNA estável e convergente. Uma RNA representativa deve mapear a saída para qualquer valor do conjunto de entrada. A estabilidade garante o resultado mas não garante a exatidão.

Uma RNA convergente produz relações de entrada/saída exatas. Então, a convergência está relacionada com a exatidão dos resultados da rede. A magnitude do erro entre os valores reais e os preditos pela rede é a medida direta da convergência.

2.9.1 Dinâmica de treinamento

A operação da rede neural constitui-se de 3 etapas:

- 1 treinamento - ajuste dos parâmetros do modelo
- 2 teste - validação dos parâmetros do modelo
- 3 produção - utilização do modelo.

Na etapa do treinamento é escolhido o algoritmo de aprendizado juntamente com os parâmetros de aprendizado. O aprendizado é o processo pelo qual a rede adapta seus parâmetros

de forma a satisfazer os requisitos de mapeamento estabelecidos. Quanto à dinâmica de atualização dos parâmetros de treinamento, pode-se citar:

Batch - os parâmetros são ajustados somente ao final de cada “*epoch*” - processamento de todo o conjunto de observações, ou seja, a cada “*epoch*” os parâmetros da rede são ajustados. Nesta dinâmica, o treinamento é menos influenciado pela ordem de apresentação dos padrões, é menos suscetível a oscilações, porém a velocidade de aprendizado (convergência) geralmente é mais baixa.

Incremental - os parâmetros são ajustados ao final do processamento de cada observação, ou seja a cada observação ajusta-se os parâmetros da rede. Nesta dinâmica, a ordem da apresentação dos padrões é importante para a velocidade de aprendizado da rede e, em alguns casos, deve-se reorganizar esta ordem, de forma a acelerar o treinamento.

2.9.2 Tipos de Aprendizado

Existem muitas maneiras de se treinar uma RNA, a maioria se divide em dois grupos:

- **Aprendizado supervisionado** - utiliza os dados de entrada especificados e a saída desejada. Se a saída gerada pela rede for diferente da saída desejada, a rede deverá ajustar seus pesos de forma a reduzir o erro. Essa dinâmica é repetida para todo conjunto de dados (entrada/saída) inúmeras vezes, até que a taxa de erro esteja dentro de uma faixa considerada satisfatória.

O processo básico do aprendizado supervisionado pode ser resumido pelos seguintes passos (para um treinamento incremental):

Passo 1: Escolha inicial dos pesos, em geral esses valores são pequenos e escolhidos aleatoriamente;

Passo 2: Apresentação de uma entrada cuja saída correspondente é conhecida;

Passo 3: Cálculo da saída a ser gerada pela rede;

Passo 4: Cálculo do erro;

Passo 5: Ajuste dos pesos;

Passo 6: Verificação do erro, se satisfatório parar senão retorna ao passo 2.

- **Aprendizado não supervisionado** – este tipo de aprendizado também é conhecido como aprendizado auto-supervisionado, e classifica os padrões similares sem utilizar pares (entrada/saída), isto é, no treinamento da rede são usados apenas valores de entrada. A rede trabalha essas entradas e se organiza de modo a classificá-las mediante algum critério de semelhança. Esse tipo de rede utiliza os neurônios como classificadores, e os dados de entrada como elementos de classificação.

2.9.3 Alguns Tipos de Aprendizado

Aprendizado por Correção do Erro: é o tipo mais comum de aprendizado usado em rede atualmente. É uma forma de aprendizado supervisionado, onde se ajusta os pesos e proporção a saída do vetor erro, $\underline{\epsilon}$. Sua saída tem n componentes, onde n é o número de nodos na saída da camada. Vamos denominar de ϵ_n o n -ésimo componente do vetor erro, e obteremos um componente erro para cada saída da rede.

Definindo o erro da saída para um nóduo simples na camada de saída:

$$\epsilon_n = d_n - c_n \quad (2.8)$$

onde ϵ_n é o erro da saída, d_n é a saída desejada, e c_n é a saída calculada, somente para o n -ésimo neurônio da camada de saída. Então calcula-se o erro quadrático total da camada de saída, E , como:

$$E = \sum_n \epsilon_n^2 = \sum_n (d_n - c_n)^2 \quad (2.9)$$

Conhecendo E , podemos calcular a mudança no peso para a i -ésima conexão para o j -enésimo neurônio, então w_{ij} fica:

$$\Delta w_{ij} = \beta_j a_i E \quad (2.10)$$

β_j é uma constante de proporcionalidade para o neurônio j (tipicamente, $0 < \beta_j \ll 1$), e a_i é a i -ésima entrada do neurônio j .

Aprendizado reforçado: é um tipo de aprendizado supervisionado do tipo correção do erro. No Aprendizado por Correção do Erro, calculamos um vetor com os valores dos erros $\underline{\epsilon}_n$. Deste modo, n diferentes valores representam o desempenho da rede. Em contraste, o

Aprendizado Reforçado tem somente um erro escalar representando o desempenho da rede. Como só temos um valor de erro para calcular, o aprendizado reforçado é mais fácil e simples do que o Aprendizado por Correção do Erro. O aprendizado Reforçado é “Seletivamente Supervisionado”, e requer um baixo nível de informação, possivelmente em intervalos infreqüentes.

Aprendizado Estocástico (Randômico): utiliza estatística, probabilidade e ou processos randômicos para ajustar os pesos. Uma mudança nos pesos é aceita se reduz o vetor erro $\underline{\epsilon}_n$. Se a mudança aumenta o erro, geralmente a mudança é rejeitada. Deste modo, uma mudança pode ser aceita se estiver enquadrada na análise de probabilidade, se tem uma melhor probabilidade de mover para um mínimo de erro. Aparentemente aceitando pesos menores o processo estocástico escapa de um mínimo local e se move para um mínimo global.

RNA's de Pesos Fixos "Hardwired": tem todas as conexões e pesos pré-determinados. Tem vantagem na velocidade, e podem ser usados a priori em reconhecimento de voz, processamento de linguagem, visão e robótica.

Aprendizado Hebbian: o ajuste dos pesos é baseado em uma correlação entre os dois nódulos e o peso associado a ele. O Aprendizado Hebbian usa uma proporcionalidade direta. Como o nódulo a_i é conectado ao nódulo b_j da próxima camada, o ajuste do peso w_{ij} é feito de acordo com a equação:

$$w_{ij,\text{novo}} = w_{ij,\text{velho}} + \beta_j a_i b_j \quad (2.11)$$

onde β_j é a taxa constante de aprendizado para o nódulo j , e $0 < \beta_j < 1$.

2.9.3.1 Aprendizado por Retropropagação (“Backpropagation”)

Vários modelos de RNA's foram desenvolvidas desde 1943. O modelo de algoritmo mais difundido, e talvez o mais utilizado, é o “backpropagation” (BP), pertencente à classe das RNA's não-recorrentes. Para que este tipo de aprendizado pudesse não se restringir somente às redes compostas por duas camadas (uma entrada e uma saída) foram criadas uma técnica chamada de retropropagação, “backpropagation”, para redes com conexões intercamadas com alimentação direta (perceptron).

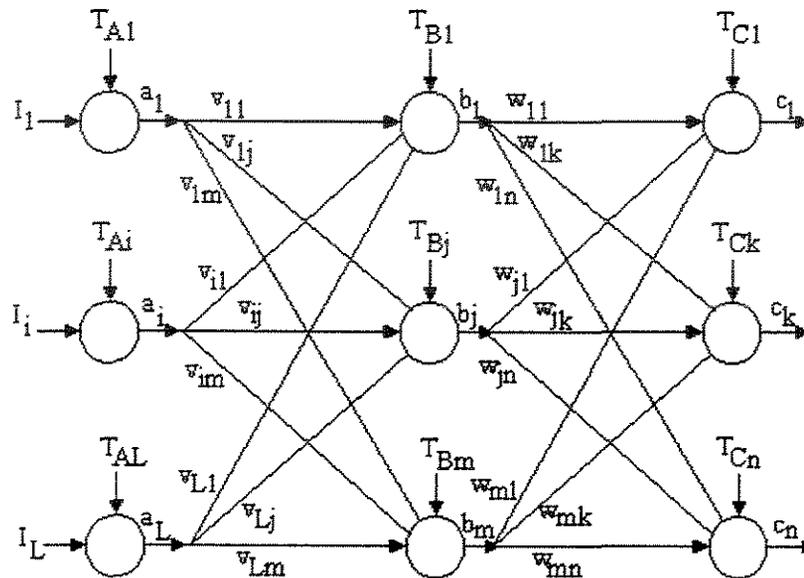


Figura 2.7 – RNA do tipo Perceptron com três camadas.

O “backpropagation” aprende mapeando a entrada com as saídas desejadas pela minimização da função erro. Tipicamente usa-se a soma quadrática dos erros. O componente de saída do vetor do n-ésimo nóculo da camada de saída é calculado, $\epsilon_n = d_n - c_n$, onde d_n é o valor de saída desejado e c_n é o valor de saída calculado. A soma quadrática dos erros, E, é:

$$E = \sum \epsilon_n^2 = \sum (d_n - c_n)^2 \quad (2.12)$$

Ajustam-se os pesos das interconexões, v_{ij} 's e w_{jk} 's mostrados na figura 2.7 para minimizar o erro, E. A seguir temos uma descrição detalhada do algoritmo “backpropagation” (Simpson, 1990). Para usar este algoritmo, temos que ter um conjunto de dados de entrada, I_L , e saída, c_n .

Passo 1: Iniciar a Rede com valores randômicos para os pesos (v_{ij} 's e w_{jk} 's) entre os valores $[-1, 1]$. Assumir o valor residual interno (T_{Ai} , T_{Bj} , T_{Ck}) para todos os nódulos entre os valores $[-1, 1]$. Note que $i=1, 2, \dots, L$ onde L é o número de nódulos na primeira camada; $j=1, 2, \dots, m$ onde m é o número de nódulos na camada intermediária; e $k=1, 2, \dots, n$, onde n é o número de neurônios na camada de saída.

Passo 2: Introduzir o vetor de entrada I_i , calcular a saída para a primeira camada, usando a função padrão sigmoial:

$$\begin{aligned} x_i &= I_i - T_{Ai} \\ a_i &= f(x_i) = \frac{1}{1 + e^{-x_i}} \end{aligned} \quad (2.13)$$

onde, I_i é a entrada no i -ésimo nódulo na camada de entrada, T_{Ai} é o valor residual do nódulo, e a_i é a saída do nódulo.

Passo 3: Conhecida a saída da camada de entrada, calcular a saída da camada intermediária, usando a equação:

$$b_j = f\left(\sum_{i=1}^L (v_{ij} a_i) - T_{Bj}\right) \quad (2.14)$$

onde $f()$ é a função sigmoial.

Passo 4: Conhecida a saída da segunda camada, calcular a saída da última camada, usando a equação:

$$c_k = f\left(\sum_{j=1}^M (w_{jk} b_j) - T_{Ck}\right) \quad (2.15)$$

Passo 5: Agora deve-se Retropropagar o erro através da rede, começando na saída indo em direção à entrada, calcular o k -ésimo componente da saída do vetor erro, ϵ_k , para cada nódulo da última camada, de acordo com a seguinte equação:

$$\epsilon_k = c_k (1 - c_k) (d_k - c_k) \quad (2.16)$$

onde d_k é o resultado desejado e c_k é o resultado atual.

Passo 6: Continue retropropagando, movendo para a camada intermediária, calcule o j -ésimo componente do vetor erro e_j , da camada intermediária relativa a cada ϵ_k , usando a equação:

$$e_j = b_j \left(1 - b_j \right) \left(\sum_{k=1}^n (w_{jk} \epsilon_k) \right) \quad (2.17)$$

Passo 7: Ajustar os pesos, calculando o novo w_{jk} ($w_{jk,novo}$) através da equação:

$$w_{jk,novo} = w_{jk,velho} + \beta_c b_j \epsilon_k \quad (2.18)$$

para $j=1$ até m e $k=1$ até n , e o termo β_c é uma constante positiva que controla a taxa de aprendizado na última camada.

Passo 8: Ajustar os valores residuais T_{ck} ($k=1$ até n) na camada de saída, de acordo com a equação:

$$T_{ck,novo} = T_{ck,velho} + \beta_c \epsilon_k \quad (2.19)$$

Passo 9: Ajustar os pesos v_{ij} , de acordo com a equação:

$$v_{ij,novo} = v_{ij,velho} + \beta_B a_i e_j \quad (2.20)$$

para $i=1$ até L e $j=1$ até m , o termo β_B é uma constante positiva que controla a taxa de aprendizado na camada intermediária.

Passo 10: Ajustar os valores residuais T_{Bj} ($j=1$ até m) na camada intermediária, de acordo com a equação:

$$T_{Bj,novo} = T_{Bj,velho} + \beta_B e_j \quad (2.21)$$

Passo 11: Repita os passos de 2 até 10 até que o erro quadrático, E , para o vetor de saída, $\underline{\epsilon}$, seja zero ou suficientemente pequeno.

Não podemos ajustar os valores residuais da camada de entrada, dependendo do problema a ser resolvido, podemos assumir T_{Ai} igual a zero. Também nesta rede, a função sigmoidal restringe a saída entre zero e 1. Esta restrição pode causar alguns problemas quando é usada para modelagem empírica. Como a saída está no intervalo $[0, 1]$ é necessário que se faça a normalização dos dados.

O algoritmo backpropagation é uma técnica de aprendizagem do tipo gradiente descendente. Usou-se o método de Newton para minimizar o erro. A vantagem deste método é que os pesos são estimados sistematicamente mais apropriadamente do que arbitrariamente. A função da camada de saída é uma sigmoideal:

$$f(x_k) = \frac{1}{1 + e^{-x_k}} \quad (2.22)$$

A derivada parcial da função é:

$$\frac{\partial f}{\partial x_k} = \frac{e^{-x_k}}{(1 + e^{-x_k})^2} \quad (2.23)$$

Evidenciando fica:

$$\frac{\partial f}{\partial x_k} = \frac{1}{1 + e^{-x_k}} \left(1 - \frac{1}{1 + e^{-x_k}} \right) \quad (2.24)$$

Ficando:

$$\frac{\partial f}{\partial x_k} = c_k (1 - c_k) \quad (2.25)$$

O termo $c_k(1-c_k)$ no passo 5 é um gradiente do método de Newton.

Uma dificuldade com os algoritmos “backpropagation” é o tempo requerido para treinar a rede. Dependendo do tamanho da RNA, o treinamento pode levar horas ou dias. Pesquisadores têm investigado diversos procedimentos alternativos numa tentativa de agilizar o processo de aprendizagem e uma destas técnicas é o Aprendizado por Gradiente Descendente.

2.9.3.2 Treinamento pela Regra de Delta Generalizado

O método conhecido por Regra de Delta Generalizado (GDR) é um método iterativo por gradiente descendente, que minimiza o quadrado do erro e se utiliza de uma técnica conhecida por “momentum” para agilizar o processo de treinamento. “Momentum” é um peso extra adicionado aos fatores ponderais, enquanto estes são ajustados. Acelerando a variação destes fatores, aumenta-se a velocidade de treinamento.

Além da utilização da técnica chamada de “momentum”, o GDR se diferencia de outros algoritmos de retropropagação pela presença da função “bias” ao invés de valores

internos residuais de atividade (“Threshold”). Estes valores residuais internos (T_{1i} , T_{2j} , T_{3k}) se transformam na função “bias” quando são adicionados, ao invés de subtraídos, ao somatório nodal. Quando usados como função “bias” os valores de T não devem ser atualizados durante o processo de treinamento e, no GDR, devem assumir os seguintes valores fixos: $T_{1i}=0$, e $T_{2j}=T_{3k}=1$. Segue abaixo uma descrição detalhada do algoritmo Regra de Delta Generalizado (GDR).

Passo 1: Assumir randomicamente valores entre 0 e 1 para os pesos v_{ij} e w_{jk} . Para o GDR os valores residuais internos devem assumir as seguintes regras: o residual da camada de entrada deve ser igual a zero, $T_{1i}=0$, todo residual das camadas intermediárias e da camada de saída devem ser iguais à unidade, $T_{2j}=T_{3k}=1$.

Passo 2: Introduzir o vetor de entrada I_i , calcular a saída para a primeira camada, usando as equações:

$$\begin{aligned} x_i &= I_i - T_{1i} = I_i - 0 = I_i \\ a_i &= f(x_i) = \frac{1}{1 + e^{-x_i}} \end{aligned} \quad (2.26)$$

Passo 3: Conhecida a saída da camada de entrada, calcular a saída da camada intermediária, b_j , usando a equação:

$$b_j = f\left(\sum_{i=1}^L (v_{ij} a_i) + T_{2j}\right) \quad (2.27)$$

onde $f(\)$ é a função sigmoideal e $T_{2j}=1$ está sendo adicionado ao somatório (ao invés de subtraído), agindo como função “bias”.

Passo 4: Conhecida a saída da segunda camada, calcular a saída da última camada, usando a equação:

$$c_k = f\left(\sum_{j=1}^M (w_{jk} b_j) + T_{3k}\right) \quad (2.28)$$

onde $f(\)$ é a função sigmoideal e $T_{3k}=1$.

Passo 5: Continuar as etapas de 1 a 4 para o P número de pares de vetores entrada/saída. Calcular o erro quadrático total, ϵ , de acordo com as seguintes equações:

$$\varepsilon = \sum_{p=1}^P \sum_{k=1}^N (d_k^p - c_k^p)^2 \quad (2.29)$$

onde N é o número de nódulos da camada de saída, d_k^p é o valor desejado (conhecido) da saída do k -ésimo nódulo do p -ésimo treinamento da RNA.

Passo 6: Calcular o termo de declínio do gradiente, δ_{3k}^p , do k -ésimo neurônio da camada de saída para o p -ésimo par de treinamento da rede. Usando a seguinte equação:

$$\delta_{3k}^p = (d_k^p - c_k^p) \frac{\partial f}{\partial x_{3k}} \quad (2.30)$$

onde f é a função sigmoidal, e sua derivada em relação a x_{3k} é dada por:

$$\frac{\partial f}{\partial x_{3k}} = \frac{e^{-x_{3k}}}{(1 + e^{-x_{3k}})^2} \quad (2.31)$$

Deve-se notar que x_{3k} é a soma das entradas ponderadas do k -ésimo neurônio da camada de saída, ou seja, para a p -ésima sessão de treinamento tem-se:

$$x_{3k}^p = \sum_j w_{jk}^p b_j^p + T_{3k}^p \quad (2.32)$$

Passo 7: Calcular o termo de declínio do gradiente, δ_{2j}^p , do j -ésimo neurônio da camada escondida, usando a seguinte equação:

$$\delta_{2j}^p = \left(\sum_k \delta_{3k}^p w_{jk}^p \right) \frac{\partial f}{\partial x_{2j}} \quad (2.33)$$

onde o subscrito k indica o nódulo da camada de saída; x_{2j} é definido por:

$$x_{2j}^p = \sum_i v_{ij}^p a_i^p + T_{2j}^p \quad (2.34)$$

e a derivada parcial da função sigmoidal em relação a x_{2j} é dada por:

$$\frac{\partial f}{\partial x_{2j}} = \frac{e^{-x_{2j}}}{(1 + e^{-x_{2j}})^2} \quad (2.35)$$

Passo 8: Conhecendo δ_{2j}^p para a camada escondida e δ_{3k}^p para a camada de saída, calcular as variações dos fatores ponderais usando as equações abaixo:

$$\begin{aligned}\Delta v_{ij}^p &= \eta \delta_{2j}^p a_i^p + \alpha \Delta v_{ij}^{p-1} \\ \Delta w_{jk}^p &= \eta \delta_{3k}^p b_j^p + \alpha \Delta w_{jk}^{p-1}\end{aligned}\tag{2.36}$$

onde η é a taxa de aprendizado e α é o coeficiente do “momentum”. Como mencionado anteriormente, “momentum” é simplesmente um peso extra usado para agilizar a etapa de treinamento e tem seu valor normalmente restrito entre 0 e 1. Assim sendo, os termos de “momentum” $\alpha \Delta v_{ij}^{p-1}$ e $\alpha \Delta w_{jk}^{p-1}$, são as frações de variação do fator ponderal sofrida na iteração anterior.

Passo 9: Atualizar os pesos de acordo com as equações:

$$\begin{aligned}w_{jk}^p &= w_{jk}^{p-1} + \Delta w_{jk}^p \\ v_{ij}^p &= v_{ij}^{p-1} + \Delta v_{ij}^p\end{aligned}\tag{2.37}$$

onde v_{ij}^p é o fator ponderal da conexão entre o i -ésimo elemento da camada de entrada e o j -ésimo elemento da camada escondida e w_{jk}^p é o peso da conexão entre o j -ésimo elemento da camada escondida e o k -ésimo elemento da camada de saída, ambos na iteração p .

Passo 10: Repetir os passos 2 a 9 para todos os pares de entrada/saída para treinamento até que o erro quadrático seja zero ou suficientemente baixo.

Das etapas descritas acima, pode-se verificar que o GDR primeiramente calcula o erro de saída e então gera novos valores para os fatores ponderais baseado neste erro e no gradiente da função sigmoïdal. O algoritmo atualiza os nódulos de saída e propaga de volta para as camadas mais próximas até encontrar a camada de entrada. Deve-se notar que à medida que o erro diminui, a taxa de atualização dos pesos se torna pequena e portanto um grande número de iterações se torna necessário para conseguir uma pequena redução do erro de predição da rede.

Algumas modificações têm sido propostas para melhorar a convergência dos métodos de retropropagação. Um exemplo é o “Backpropagation” com Gradiente Conjugado, onde a taxa de aprendizado e o coeficiente de “momentum” são otimizados a cada passo do algoritmo e a atualização dos pesos é feita de uma única vez, depois que todas as variações de pesos são

calculadas (ao invés de ser feita para cada vetor de entrada/saída lido). Outras modificações sugerem simplesmente que estes parâmetros variem através de regras heurísticas simples, de acordo com o aumento ou diminuição do erro quadrático médio.

Porém, como os métodos por retropropagação não apresentam nenhum tipo de cálculo matricial, além de não apresentarem problemas quanto à memória requerida, a velocidade de cálculo em cada iteração é bastante elevada. No entanto, a taxa de convergência é bem menor que a dos métodos de Newton que, por serem de segunda ordem, apresentam convergência quadrática próxima ao mínimo. Estes últimos apresentam como limitação o grande espaço de memória requerido e o volume de cálculos matriciais envolvido.

Com o intuito de reduzir os problemas dos métodos de Newton, diversos métodos quasi-Newton têm sido propostos baseados em simplificações destes. A seguir descreve-se o Método quasi-Newton de Levenberg-Marquardt, que dependendo do tamanho da RNA se mostra mais eficiente que o GDR.

2.9.3.3 Algoritmo de Treinamento Levenberg-Marquardt

Os métodos quasi-Newton mostram-se em princípio mais rápidos e mais robustos do que aqueles baseados no gradiente descendente. Com base nisto, Hagan e Menhaj (1994) utilizaram o algoritmo de Levenberg-Marquardt para o treinamento de RNA's do tipo "Feedforward" e verificaram seu melhor desempenho para os casos onde existiam apenas algumas centenas de dados entrada/saída. Este método consiste numa aproximação do método de Newton e propõe que a atualização dos pesos seja dada pela seguinte equação:

$$\Delta \bar{y} = [J^T(\bar{y})J(\bar{y}) + \mu.I]^{-1} J^T(\bar{y})e(\bar{y}) \quad (2.38)$$

sendo:

$$\bar{y} = [w_{1,1}^1, w_{1,2}^1, \dots, w_{SI,R}^1, b_1^1, \dots, b_{SI}^1, w_{1,1}^2, \dots, w_{i,j}^k, \dots, b_{S,k}^k, \Lambda, b_{SM}^M], \text{ vetor peso;} \quad (2.39)$$

$$J(\bar{y}) = \begin{bmatrix} \frac{\partial e_1(\bar{y})}{\partial y_1} & \frac{\partial e_1(\bar{y})}{\partial y_2} & \dots & \frac{\partial e_1(\bar{y})}{\partial y_n} \\ \frac{\partial e_2(\bar{y})}{\partial y_1} & \frac{\partial e_2(\bar{y})}{\partial y_2} & \dots & \frac{\partial e_2(\bar{y})}{\partial y_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{NP}(\bar{y})}{\partial y_1} & \frac{\partial e_{NP}(\bar{y})}{\partial y_2} & \dots & \frac{\partial e_{NP}(\bar{y})}{\partial y_n} \end{bmatrix}, \text{ matriz jacobiana dos erros;} \quad (2.40)$$

$$\frac{\partial e_q(\bar{y})}{\partial w_{i,j}^k} = \delta_i^k \cdot a_j^{k-1} \quad \text{e} \quad \frac{\partial e_q(\bar{y})}{\partial b_i^k} = \delta_i^k; \quad (2.41)$$

$q=1, 2, \dots, Np$; $k=1, 2, \dots, M-1$; $i=1, 2, \dots, Sk+1$; $j=1, 2, \dots, Sk$.

As etapas realizadas no treinamento são as mesmas já descritas. Além disso, os cálculos dos termos de declínio (δ) usam as mesmas equações do método de retropropagação, com uma simples modificação:

$$\bar{\delta} = F(\bar{x}^M) \quad (2.42)$$

Como se pode notar o cálculo da matriz jacobiana dos erros entre os valores reais e os preditos pela RNA é a etapa que mais exige esforço computacional. Para problemas com grandes quantidades de dados a memória requerida para tal processamento torna-se um fator limitante na utilização do algoritmo.

2.10 Generalização

Um dos problemas que ocorre durante o treinamento é o sobreajuste ('overfitting'), que é quando o erro durante o treinamento chega ao mínimo mas quando novos dados são apresentados o erro é grande. A rede memorizou os dados de treinamento e não generalizou para novas situações. Existem dois métodos para melhorar a generalização dos dados : Regularização e a Parada Antecipada.

2.10.1 Regularização

A regularização envolve a modificação da função a qual é normalmente a soma dos erros quadráticos do conjunto de treinamento.

A função típica usada para treinar redes neurais retroalimentadas é a soma dos erros quadráticos:

$$F = \frac{1}{N} \sum_{i=1}^N e^2 = \frac{1}{N} \sum_{i=1}^N (d_i - c_i)^2 \quad (2.43)$$

Para melhorar a generalização modifica-se a função, um termo que consiste na média da soma dos quadrados dos pesos e bias:

$$F_{\text{reg}} = \alpha F + \beta F_w \quad \text{onde } \alpha \text{ e } \beta \text{ são parâmetros da função objetivo,} \quad (2.44)$$

$$F_w = \frac{1}{n} \sum_{j=1}^n w_j^2 \quad (2.45)$$

Usando-se esta função os pesos e bias têm valores pequenos, isto força a rede a responder suavemente e então o sobreajuste é pouco provável (Foresee, 1997).

O problema da regularização é a dificuldade em determinar um valor adequado para a taxa. Se o valor for muito grande poderá levar a um sobreajuste e se for muito pequeno a rede poderá não ajustar adequadamente os dados de treinamento.

É desejável determinar esses parâmetros de uma forma automatizada, uma destas abordagens é um processo que usa a estrutura Bayesiana (MacKay, 1992).

Na estrutura Bayesiana, os pesos são considerados aleatórios e variáveis. Depois os dados são tomados, e a função densidade para os pesos podem ser antecipados de acordo com a regra de Bayes:

$$P(w|D, \alpha, \beta, m) = \frac{P(D|w, \beta, M)P(w|\alpha, M)}{P(D|\alpha, \beta, M)} \quad (2.46)$$

onde D representa o conjunto de dados, M é o modelo de rede neural usado, e w é o vetor peso da rede. $P(w|\alpha, M)$ é a densidade anterior, o qual representa o conhecimento dos pesos antes de qualquer dado ser coletado. $P(D|w, \beta, M)$ é a função de probabilidade dos dados, dados os pesos w . $P(D|\alpha, \beta, M)$ é o fator de normalização, que garante que a probabilidade total é 1.

Se assumirmos que o ruído no conjunto de treinamento é Gaussiano e a distribuição anterior para os pesos é Gaussiana, a densidade de probabilidade pode ser escrita como:

$$\begin{aligned} P(D|W, \beta, M) &= \frac{1}{Z_D(\beta)} \exp(-\beta E_D) \\ P(w|\alpha, M) &= \frac{1}{Z_w(\alpha)} \exp(-\alpha E_w) \end{aligned} \quad (2.47)$$

onde $Z_D(\beta) = (\pi/\beta)^{n/2}$ e $Z_w(\alpha) = (\pi/\alpha)^{N/2}$. Se substituirmos estas probabilidades na eq. 2.46, obtemos:

$$\begin{aligned} P(w|D, \alpha, \beta, M) &= \frac{\frac{1}{Z_w(\alpha)} \frac{1}{Z_D(\beta)} \exp(-(\beta E_D + \alpha E_w))}{\text{Fator de Normalização}} \\ &= \frac{1}{Z_F(\alpha, \beta)} \exp(-F(w)) \end{aligned} \quad (2.48)$$

Nesta estrutura Bayesiana, os pesos ótimos deverão maximizar a probabilidade posterior $P(w|D, \alpha, \beta, M)$. Maximizando a probabilidade posterior é equivalente a minimizar a função objetiva regularizada $F = \beta E_D + \alpha E_w$.

Para otimizar os parâmetros da função objetivo α e β , agora vamos considerar a aplicação da regra de Bayes. Agora temos:

$$P(\alpha, \beta|D, M) = \frac{P(D|\alpha, \beta, M)P(\alpha, \beta|M)}{P(D|M)} \quad (2.49)$$

Se assumirmos como uniforme a densidade anterior $P(\alpha, \beta|D, M)$ para os parâmetros α e β , então a maximização da posterior é realizada pela maximização da função de probabilidade $P(D|\alpha, \beta, M)$. Note que esta função de probabilidade é o fator de normalização da eq. 2.46. Desde que todas as probabilidades tenham a forma Gaussiana, podemos conhecer a forma da densidade posterior da eq. 2.46. Isto é mostrado na eq. 2.48. Agora podemos resolver a eq. 2.46 para o fator de normalização.

$$\begin{aligned}
P(D|\alpha, \beta, M) &= \frac{P(D|w, \beta, M)P(w|\alpha, M)}{P(w|D, \alpha, \beta, M)} \\
&= \frac{\left[\frac{1}{Z_D(\beta)} \exp(-\beta E_D) \right] \left[\frac{1}{Z_w(\alpha)} \exp(-\alpha E_w) \right]}{\frac{1}{Z_F(\alpha, \beta)} \exp(-F(w))} \\
&= \frac{Z_F(\alpha, \beta)}{Z_D(\beta)Z_w(\alpha)} \frac{\exp(-\beta E_D - \alpha E_w)}{\exp(-F(w))} = \frac{Z_F(\alpha, \beta)}{Z_D(\beta)Z_w(\alpha)}
\end{aligned} \tag{2.50}$$

Note que conhecemos as constantes $Z_D(\beta)$ e $Z_w(\alpha)$ da eq. 2.47, somente uma parte não é conhecida $Z_F(\alpha, \beta)$. Mas podemos estimá-la por expansão com série de Taylor. Desde que a função objetivo tenha a forma quadrática em uma pequena área ao redor do ponto mínimo, podemos expandir $F(w)$ em volta do ponto mínimo da densidade posterior w^{MP} , onde o gradiente é zero. Resolvendo para a constante de normalização obtemos:

$$Z_F \approx (2\pi)^{N/2} \left(\det(H^{MP})^{-1} \right)^{1/2} \exp(-F(w^{MP})) \tag{2.51}$$

onde $H = \beta \nabla^2 E_D + \alpha \nabla^2 E_w$ é a matriz de Hessian da função objetivo. Colocando este resultado na eq. 2.50, podemos resolver para os valores ótimos de α e β no ponto mínimo. Fazemos isto pegando a derivada em relação a cada logaritmo da eq. 2.50 e igualamos a zero. Obtemos:

$$\alpha^{MP} = \frac{\gamma}{2E_w(w^{MP})} \quad \text{e} \quad \beta^{MP} = \frac{\pi - \gamma}{2E_D(w^{MP})} \tag{2.52}$$

onde $\gamma = N - 2\alpha^{MP} \text{tr}(H^{MP})^{-1}$ é chamado de número efetivo de parâmetros, e N é o número total de parâmetros da rede. O parâmetro γ é a medida de quantos parâmetros da rede são efetivamente usados na redução da função erro. Ele pode variar de 0 até N .

A otimização Bayesiana da regularização dos parâmetros requer o cálculo da matriz de Hessian de $F(w)$ no ponto mínimo w^{MP} . Propomos o uso da aproximação de Gauss-Newton para a matriz Hessian, a qual está disponível se o algoritmo de otimização Levenberg-Marquardt é usado para a localização do ponto mínimo. Os cálculos adicionais requeridos para a otimização da regularização são mínimos.

Aqui estão os passos necessários para a otimização Bayesiana dos parâmetros da regularização, com a aproximação de Gauss-Newton para a matriz Hessian:

1. Inicializar α e β e os pesos. Escolhemos fixar $\alpha = 0$ e $\beta = 1$ e inicializar os pesos. Depois do primeiro passo do treinamento, a função objetivo irá recuperar o ajuste inicial.
2. Usar o algoritmo de Levenberg-Marquardt para minimizar a função objetivo $F = \beta E_D + \alpha E_w$.
3. Calcular o efetivo número de parâmetros $\gamma = N - 2\alpha \text{tr}(H)^{-1}$ fazendo uso da aproximação de Gauss-Newton disponível no algoritmo de treinamento Levenberg-Marquardt: $H = \nabla^2 F(w) \approx 2\beta J^T J + 2\alpha I_N$, onde J é a matriz Jacobiana do conjunto de erro do treinamento.
4. Calcular novos parâmetros estimados para a função objetivo $\alpha = \frac{\gamma}{2E_w(w)}$ e

$$\beta = \frac{n - \gamma}{2E_D(w)}.$$

5. Repetir os passos de 2 até 4 até a convergência.

Sabemos que a cada reestimação, os parâmetros da função objetivo estão mudando, por esta razão o ponto mínimo está se movendo. Se atravessarmos a superfície de desempenho movemos em direção ao próximo ponto mínimo, então as novas estimativas para a função objetivo serão mais precisas. Eventualmente, a precisão poderá ser boa o bastante que a função objetivo poderá não ter mudança significativa nas próximas iterações. Então obteremos a convergência.

Quando a aproximação de Gauss-Newton é usada para a regularização Bayesiana, melhores resultados são obtidos se o conjunto de treinamento é normalizado na faixa de $[-1,1]$ (ou alguma região similar). Normalmente, normaliza-se entrada e saída.

Depois do treinamento, existem algumas inspeções que devem ser feitas. Primeiro, se o número efetivo de parâmetros final, γ , é muito próximo do atual número de parâmetros N , então a rede poderá não ser boa o bastante para representar verdadeiramente a função. Neste

caso, é só adicionar mais neurônios à camada intermediária e retreinar. Se a rede maior tiver o mesmo γ , então a rede anterior tinha neurônios suficientes. Caso contrário, mais camadas de neurônios podem ser necessárias. A segunda inspeção é a consistência dos resultados. Se a rede é suficientemente grande, então a rede maior deverá ter valores comparáveis de γ , E_D e E_w .

2.10.2 Parada Antecipada

Um outro método para melhorar a generalização é a Parada Antecipada. Os dados são separados em três subconjuntos. O primeiro subconjunto é o conjunto de treinamento, é usado para calcular o gradiente e para atualizar os pesos e bias. O segundo subconjunto é o de validação, o erro da validação é monitorado durante o treinamento. O erro na validação decresce durante a fase inicial de treinamento, como o erro do conjunto de treinamento. Deste modo, quando a rede começar a sobreajustar os dados, o erro no conjunto de validação irá aumentar. Quando o erro na validação aumenta para um número específico de iterações o treinamento é parado, e o peso e bias e o erro mínimo da validação são retornados.

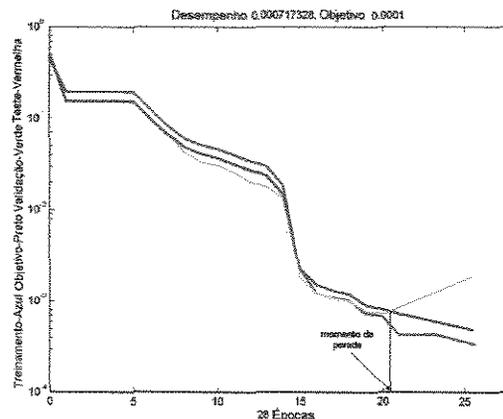


Figura 2.8 – Gráfico do treinamento com algoritmo Parada Antecipada.

O conjunto de teste não é usado durante o treinamento, mas é usado para comparar diferentes modelos. É comum graficar o erro do conjunto de teste durante o treinamento, se o erro no conjunto de teste chega ao mínimo num número diferente de iterações que o conjunto de validação, isto indica se o conjunto está bem dividido.

Tanto a Regularização quanto a Parada Antecipada podem assegurar a generalização da rede quando bem aplicados. Quando estiver usando a regularização Bayesiana, é importante treinar a rede até alcançar a convergência.

Ao se usar a Parada Antecipada deve-se ter muito cuidado de não usar um algoritmo que convirja muito rapidamente. Se estiver usando um algoritmo muito rápido deve-se determinar os parâmetros para que a convergência seja relativamente baixa.

Com a Parada Antecipada, a escolha do conjunto de validação é muito importante. O conjunto de validação deve representar todos os pontos do conjunto de treinamento.

Com ambos algoritmos, é bom treinar a rede iniciando-se de várias condições iniciais. É possível que cada método falhe em certas circunstâncias. Testando de diversas situações iniciais pode-se verificar quão robusto é o desempenho da rede.

A regularização Bayesiana generaliza melhor que a parada antecipada, quando treina-se redes para aproximar funções. Isto é porque a regularização Bayesiana não requer a separação do conjunto de validação e treinamento, usam-se todos os dados. Esta vantagem é especialmente notada quando o conjunto de dados é pequeno.

2.11 Uso de Redes Neurais em Extração Líquido-Líquido

O uso de Redes Neurais na extração é bastante recente, tanto na modelagem de sistemas como no controle de colunas de extração.

Bogdan et al (1995) usou redes para descrever o equilíbrio líquido em sistemas bifásicos aquosos da proteinase neutra e oxitetraciclano da *Streptomyces rimosus* no sistema PEG- 6000 e sulfato de amônia. Uma rede “feedforward” multicamada foi treinada aplicando-se o algoritmo “backpropagation” obtendo-se uma topologia 4-2-1.

Gibes et al (1996) modelaram a extração de terras raras com redes neurais em vários sistemas, isto mostrou que os raios dos lantanídeos poderiam ser usados para generalizar o procedimento de extração com solventes. Os dados para o treinamento foram pré-processados através de transformação logarítmica. A representação da extração foi melhorada através de camadas do tipo “self-organizing” (Kohonen) na rede neural, obtendo uma topologia 3-10-1.

Chouai et al (2000), estudaram a aplicação de redes neurais em uma coluna de anéis pulsada, na extração de uma mistura de tributilfosfato e dodecano (como solvente), onde o objetivo foi minimizar a concentração de ácido oxálico no refinado. A resposta de pH e condutividade para variações de frequências de pulsação e vazão de solvente foram inferenciadas através de um conjunto de dados experimentais. A rede foi treinada com um

conjunto de 6000 dados de entrada-saída e um conjunto de teste de 2992 dados, a topologia final obtida foi 11-9-2.

3.0 Materiais e Métodos

3.1 Software

O software usado para este trabalho foi o Matlab 5.3, mais precisamente a ferramenta (toolbox) de redes neurais.

3.2 Algoritmos

Os algoritmos usados neste trabalho foram:

- Levenberg-Marquardt
- Parada Antecipada
- Regularização Bayesiana

Estes algoritmos foram explicados com detalhes na Revisão de Literatura.

4.0 Resultados e Discussão

4.1 Dados

Os dados usados para o treinamento das redes neurais foram separados em dois grupos: dados de operação descontínuo e dados de operação contínuo.

4.2 Operação Descontínua

O conjunto de dados para a predição do coeficiente de partição é composto por 120 pares entrada-saída, com um erro máximo experimental de mais ou menos 10%. As variáveis usadas para o treinamento foram :

Tabela 4.1- Variáveis de entrada da Operação Descontínua

Entrada	Valor mínimo	Valor máximo
Massa Molar PEG	400	8000
Massa específica da fase PEG	1078,6 kg/m ³	1098,9 kg/ m ³
Viscosidade da fase PEG	5,9mPa .s	46 mPa.s
Massa específica da fase sal	1122,1kg/m ³	1292,3kg/m ³
Viscosidade da fase sal	0,9 mPa.s	2,5 mPa.s

Tabela 4.2 - Variáveis de saída da Operação Descontínua

Saída	Valor mínimo	Valor máximo
Coefficiente de partição	0,079	5,68

Dos 120 dados, 60 são dados experimentais e 60 dados foram calculados com base no erro máximo de 10% com auxílio de uma função para gerar erro aleatório de mais ou menos 10% em relação a média dos pontos, isso garante que o erro não tenha tendência e seja válido, como se fosse um dado experimental.

Na modelagem descontínua existem duas áreas uma indicando a tendência de partição para a fase PEG a outra para a fase Sal, a faixa intermediária não tem significado físico, faz parte apenas da modelagem.

4.3 Operação Contínua

O conjunto de dados para a predição da Porcentagem de Extração e da Fração de Retenção da fase dispersa é composto por 180 pares entrada-saída, com um erro máximo $\pm 10\%$. As variáveis usadas para o treinamento foram:

Tabela 4.3 - Variáveis de entrada da Operação Contínua

Entrada	Valor Mínimo	Valor Máximo
Coefficiente de Partição	0,595	0,92
Vazão Total das Fases	4,7 mL/min	10,9 mL/min
Frequência	0,1 pulsos/s	1 pulsos/s

Tabela 4.4 – Variáveis de saída da Operação Contínua

Saída	Valor Mínimo	Valor Máximo
Porcentagem de Extração	11,82	75,83
Fração de Retenção	0,0712	0,4628

Dos 180 pares de dados, 72 são dados experimentais e 108 são dados calculados a partir de aproximações e de um erro máximo de $\pm 10\%$.

Assumiu-se que a eficiência de Murphree mantém-se constante para coeficientes de partição do mesmo PEG, porque a coluna é a mesma e as condições hidrodinâmicas também são as mesmas.

Com as simplificações fica:

$$C_r = C_f - E * [C_f - C_r^*]$$

Com C_r^* que é a concentração da proteína no refinado da operação descontínua, podemos calcular o Índice de Recuperação, para mais duas linhas de amarração ampliando a faixa da modelagem.

Para as Frações de Retenção como as condições hidrodinâmicas são constantes somente a viscosidade e a densidade variam com isso fica:

$$\varphi = k_1 \exp[k_2 (A_f - |A_f|)] V_d^{0,86} (V_c + V_d)^{0,28} \Delta\rho^{-0,30} \rho d^{-0,93} \mu d^{-0,77} \alpha^{-0,56} h^{-0,56}$$

portanto:

$$\varphi = \text{cte} \cdot \Delta\rho^{-0,30} \rho d^{-0,93} \mu d^{-0,77}$$

Depois esses dados foram usados para calcular dados com desvio de mais ou menos 10% em relação a média dos pontos, para simular o erro dos experimentos.

4.4 Normalização

Normalizou-se os dados entre a faixa de 0,1 e 0,9 ao invés de 0 e 1 porque o conjunto de dados apresentava muitos dados próximos ao zero tornando a matriz esparsa e de difícil solução.

4.5 Função de Ativação

Usou-se a função de ativação sigmoideal devido a faixa de normalização que é a mesma desta função.

4.6 Análise do Treinamento

Os critérios de análise usados para os treinamentos foram: dispersão, que neste trabalho é definido como o somatório dos erros entre a saída da rede e os dados experimentais, desvio padrão dos dados, além da análise visual dos gráficos.

O melhor projeto é definido como aquele que minimiza o erro de predição (baixa dispersão, maioria dos dados na faixa de erro de $\pm 10\%$ e desvio padrão próximo aos dados experimentais) para o conjunto de dados que não foi treinado (conjunto de teste).

4.7 Divisão do Conjunto de dados

4.7.1 Descontínuo

Os 120 dados foram divididos em três conjuntos de dados para treinamento do algoritmo Parada Antecipada:

- 80 dados de treinamento
- 20 dados de validação
- 20 dados de teste

Para os demais algoritmos os dados foram divididos em dois conjuntos:

- 100 dados de treinamento
- 20 dados de teste.

4.7.2 Contínuo

Os 180 dados foram divididos em três conjuntos para treinamento do algoritmo Parada Antecipada:

- 132 dados de treinamento
- 24 dados de validação

- 24 dados de teste

Para os demais algoritmos os dados foram divididos em dois conjuntos:

- 156 dados de treinamento
- 24 dados de teste.

Existem dois conjuntos de dados: dados com variação na entrada (dados com erro experimental) e dados sem variação na entrada (que são calculados através da média dos dados no ponto).

4.8 Treinamento

4.8.1 Operação Descontínua

4.8.1.1 Seleção das Variáveis

Para verificar-se a necessidade de todas as variáveis, as redes foram treinadas com todas variáveis e depois retirando-as do modelo para verificar sua importância. O critério para exclusão das variáveis foi: se a dispersão não aumentar (ou aumentar muito pouco) com a exclusão da variável, esta não é necessária para o modelo podendo ser descartada.

Para verificação das Variáveis Operação Descontínua usou-se o algoritmo Regularização Bayesiana com excesso de neurônios, porque neste algoritmo quando o número de neurônios é suficiente, os parâmetros da rede (como as somas quadráticas dos erros e dos pesos e o número efetivo de parâmetros) tornam-se constantes mesmo aumentando o número de neurônios, então se a rede tiver um número grande de neurônios ela sempre irá representar satisfatoriamente a função usando apenas a quantidade necessária de neurônios. As funções de ativação usadas foram sigmoideal.

4.8.1.2 Variáveis

Entrada: massa molar PEG, viscosidade da fase PEG, massa específica da fase PEG viscosidade da fase Sal, massa específica da fase Sal.

Saída: Coeficiente de Partição

4.8.1.3 Treinamento para Seleção das Variáveis

Para iniciar o treinamento usou-se todas as propriedades do sistema (massa molar do PEG, massa específica e viscosidade das fases (PEG e Sal)). Treinando-se e ampliando os gráficos do coeficiente de partição nas faixas de operação para os conjuntos de treinamento e teste:

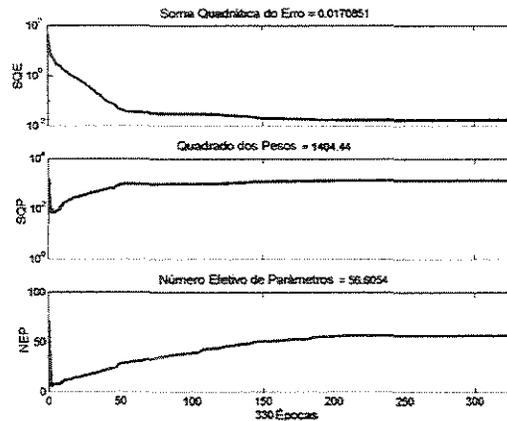


Figura 4.1- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.

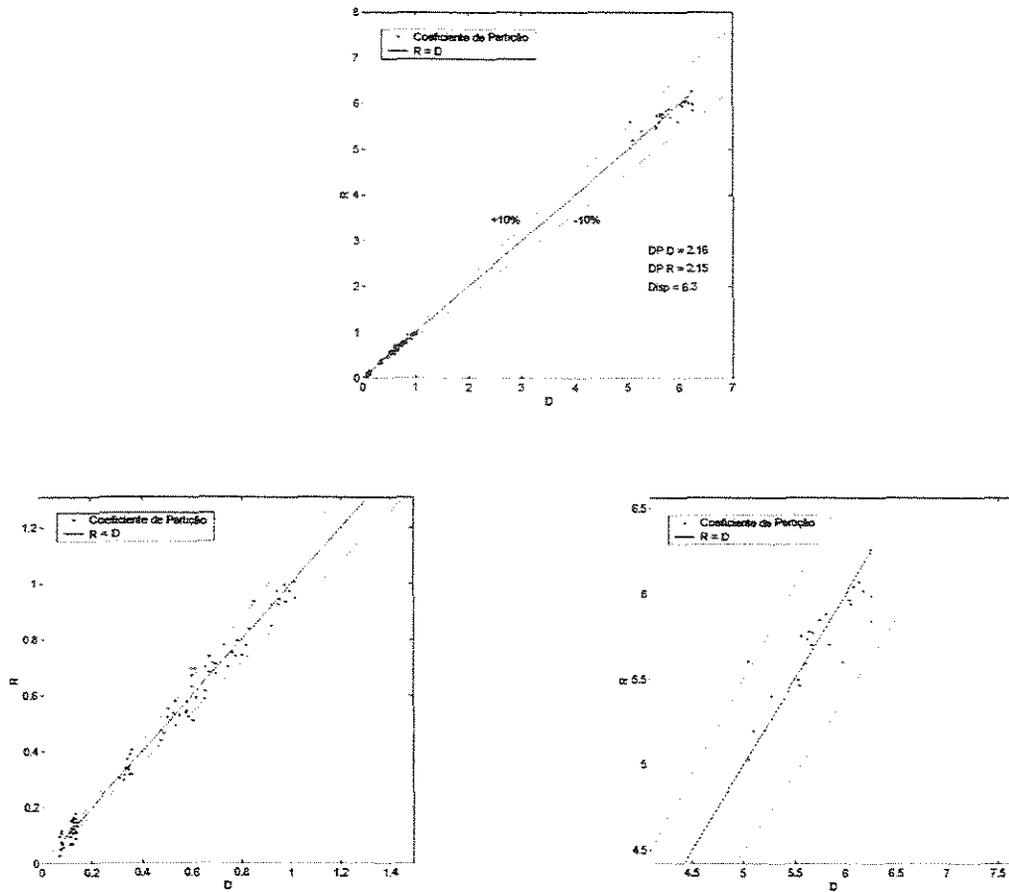


Figura 4.2 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.

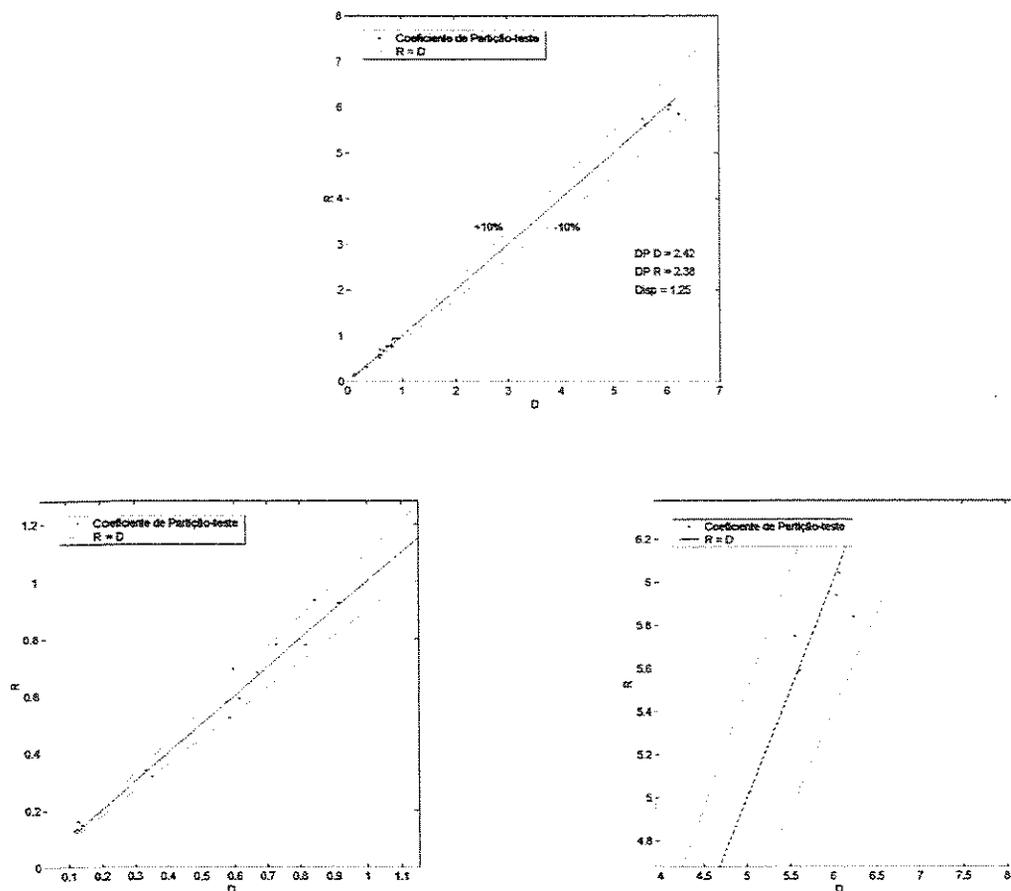


Figura 4.3 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.

Como observou-se através das figuras 4.2 e 4.3 a dispersão foi baixa e a maioria dos dados ficou na faixa de erro de $\pm 10\%$, com isso nota-se que com essas variáveis conseguimos um bom modelo, depois verificaram-se se todas as variáveis eram necessárias, retirando-as do modelo.

Treinando-se com massa molar PEG, massa específica da fase PEG, viscosidade da fase PEG, viscosidade da fase sal, obteve-se:

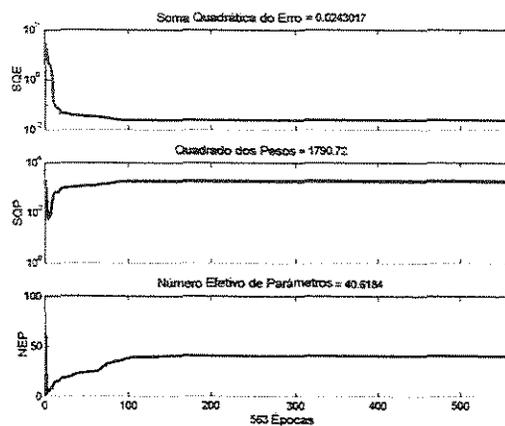


Figura 4.4- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.

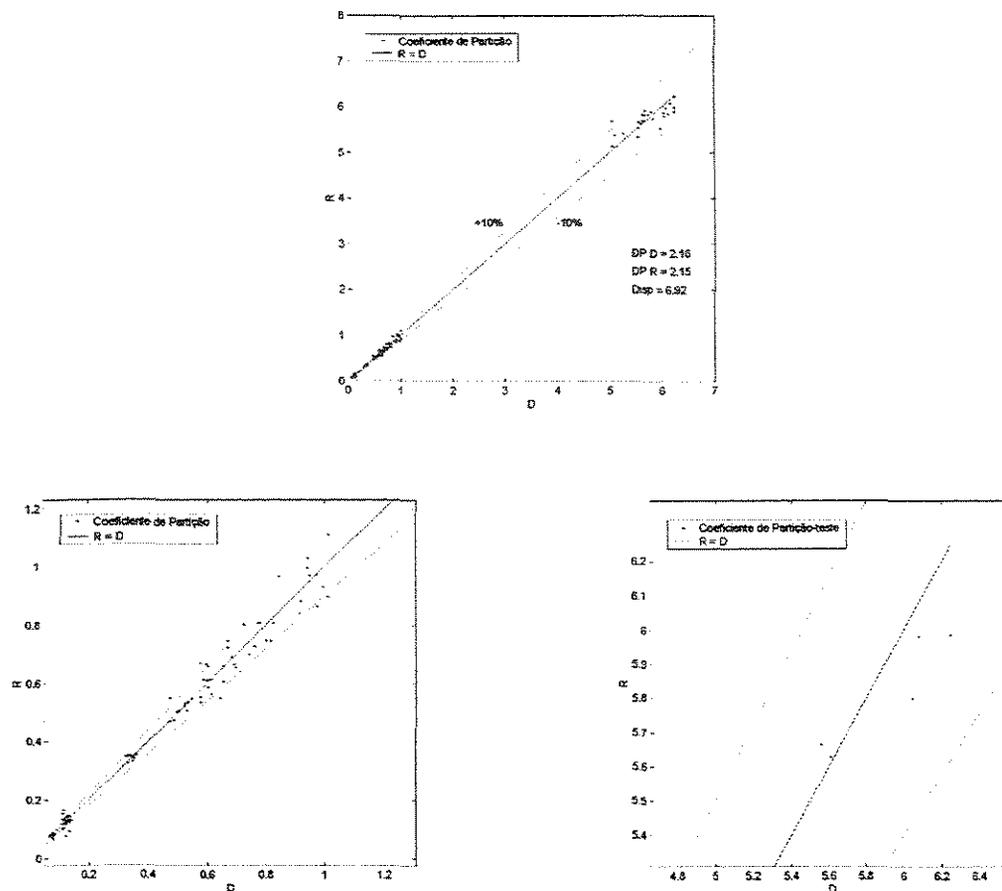


Figura 4.5- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.

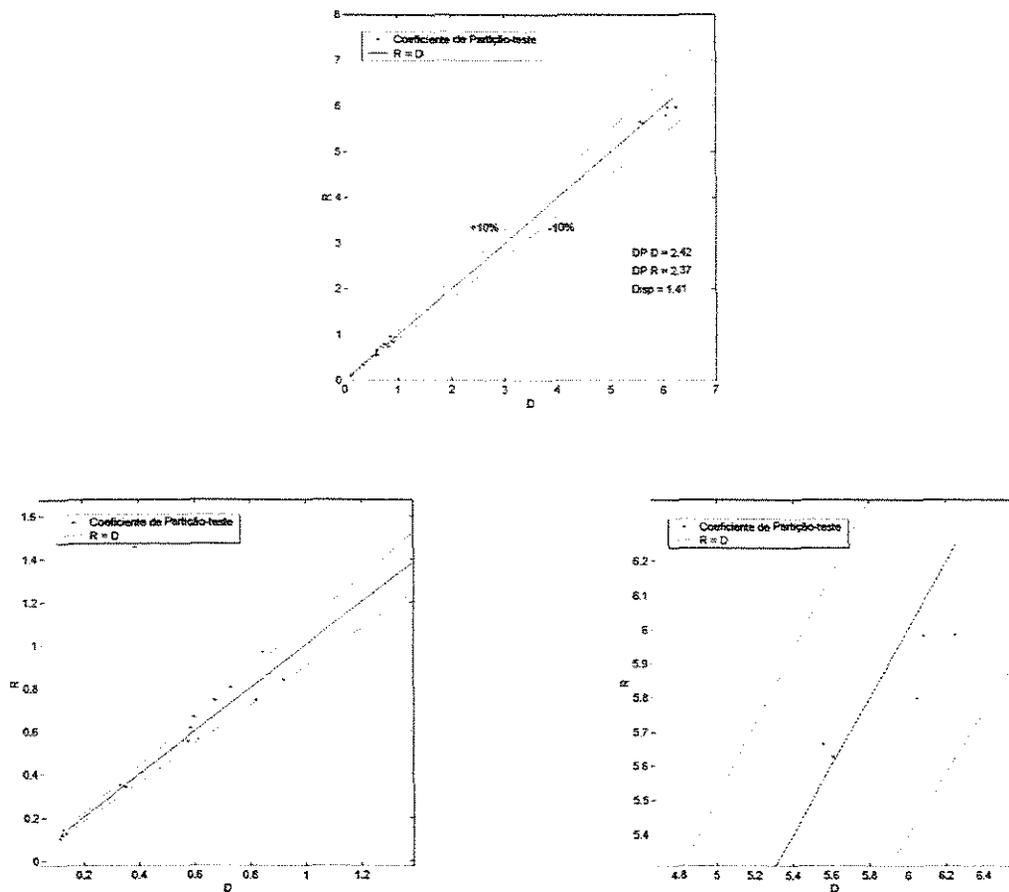


Figura 4.6- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.

Através da análise dos gráficos 4.5 e 4.6 observou-se que houve um pequeno aumento da dispersão do conjunto de teste, mas os dados não ultrapassaram a faixa de $\pm 10\%$, concluindo-se que a variável massa específica da fase Sal não é necessária.

Treinando-se com massa molar PEG, massa específica da fase PEG, viscosidade da fase PEG, massa específica da fase sal, obteve-se:

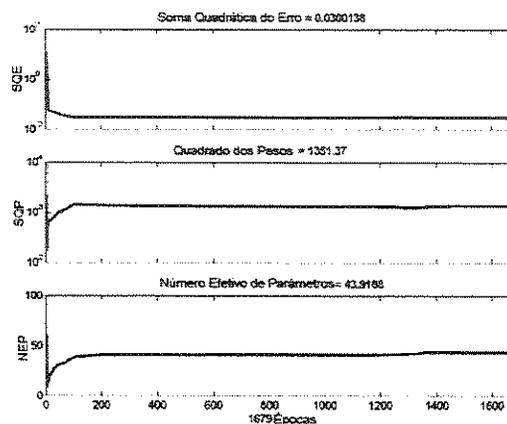


Figura 4.7- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.

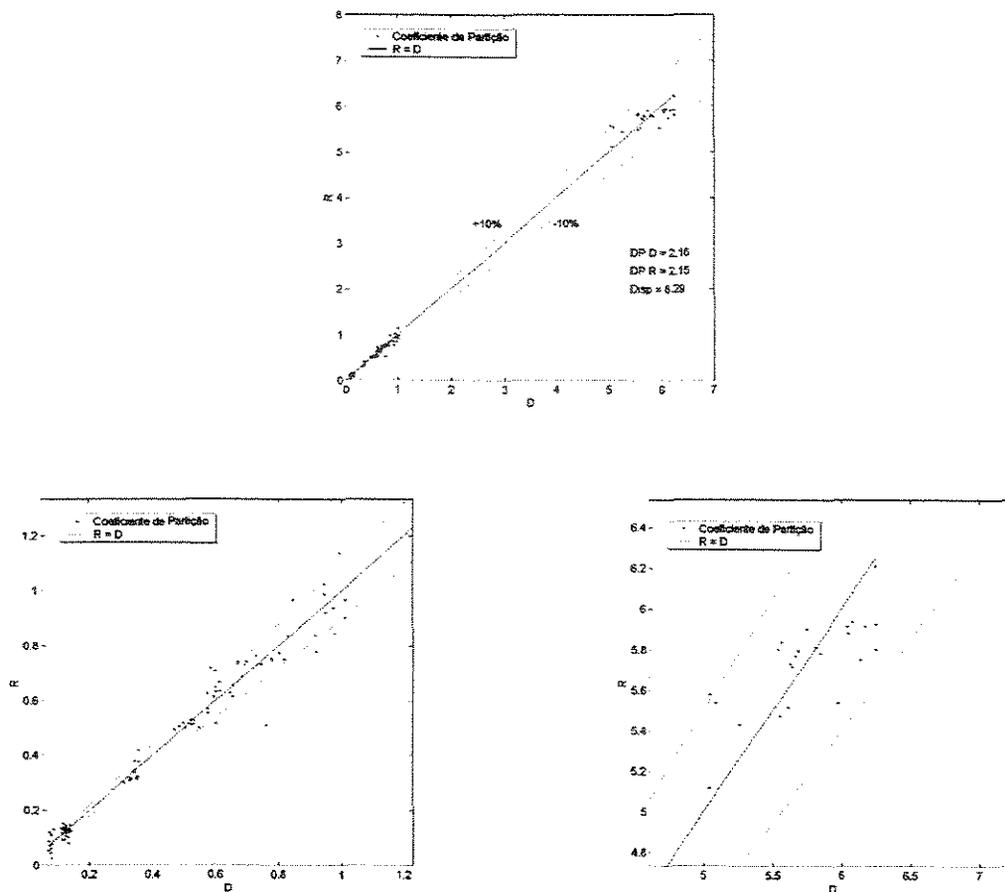


Figura 4.8- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.

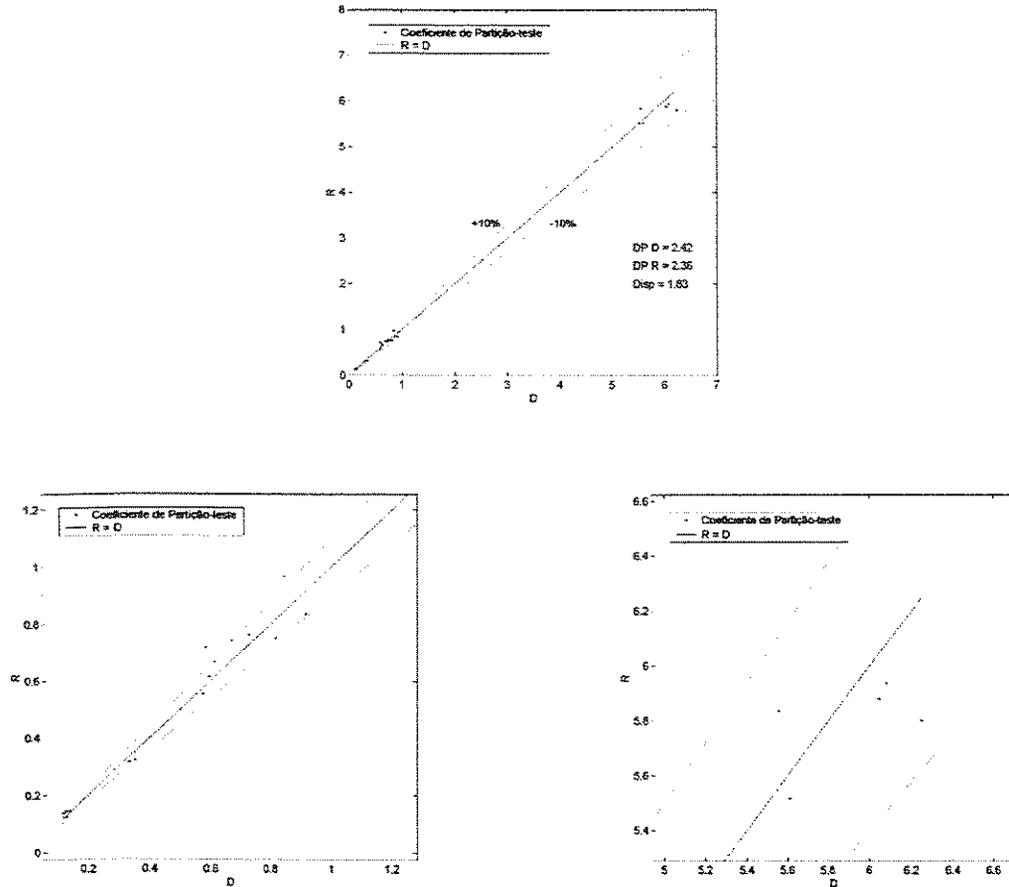


Figura 4.9 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.

Analisando-se as figuras 4.8 e 4.9 nota-se que houve um aumento na dispersão e alguns dados ultrapassaram a faixa de $\pm 10\%$, indicando que a variável viscosidade da fase sal é necessária.

Treinando-se com as variáveis massa molar PEG, massa específica da fase PEG, viscosidade da fase PEG, obteve-se:

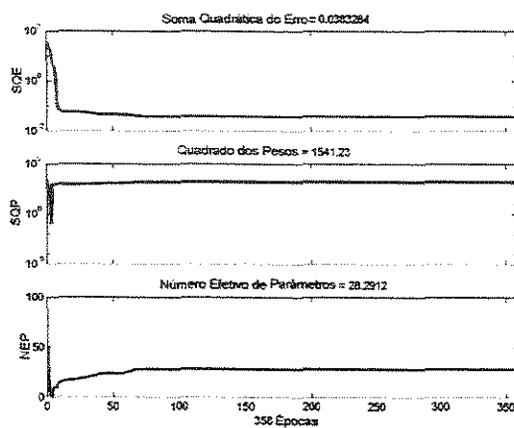


Figura 4.10- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.

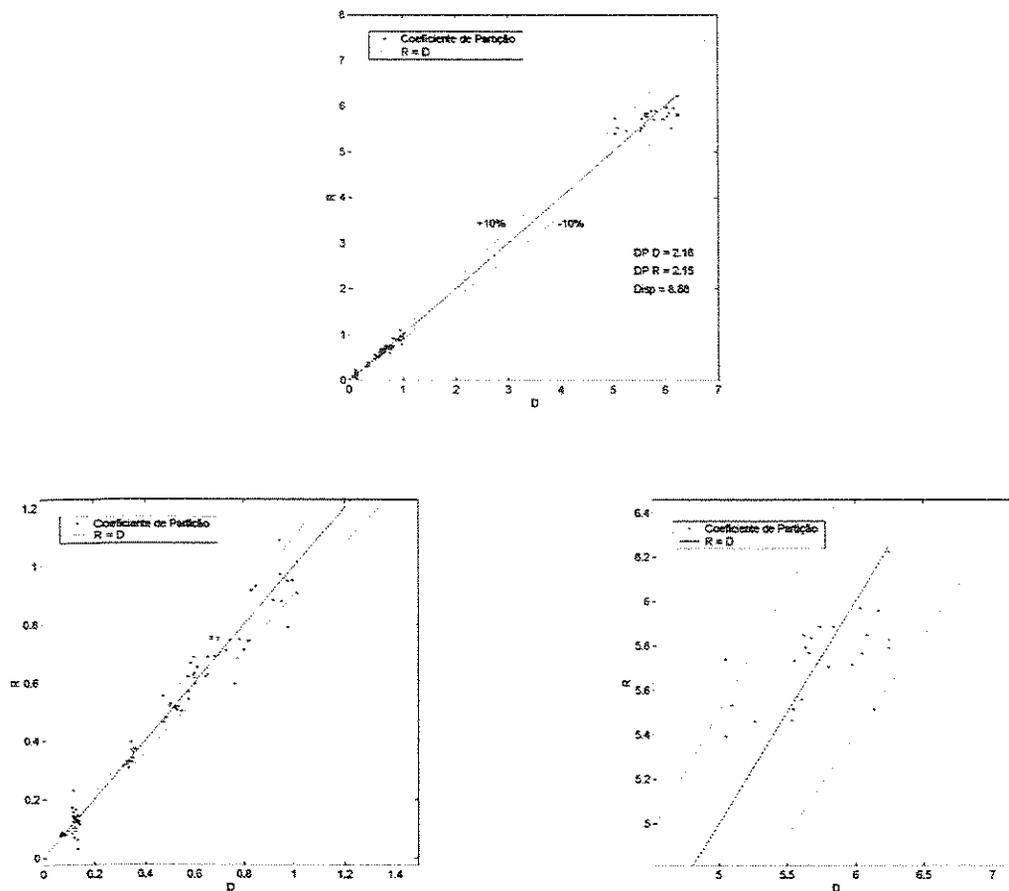


Figura 4.11 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.

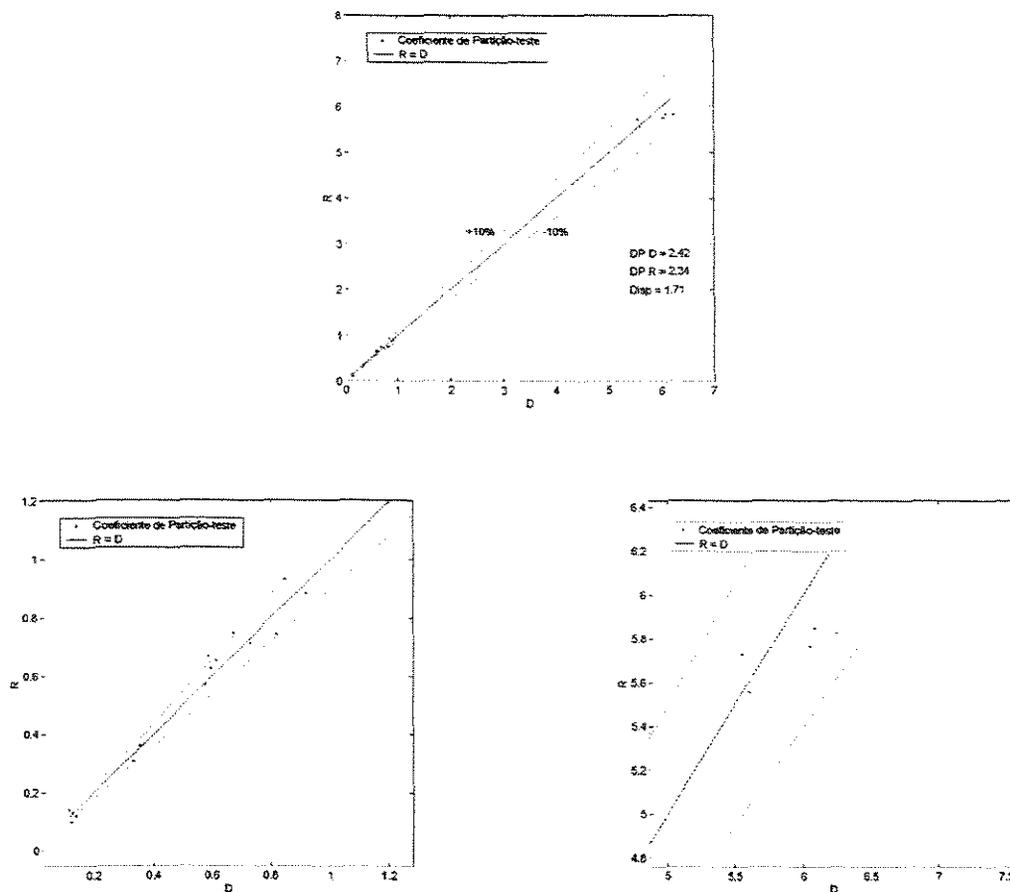


Figura 4.12 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.

Analisando-se as figuras 4.11 e 4.12 nota-se que este modelo tem uma baixa dispersão e os dados estão na faixa de $\pm 10\%$, observando-se que este é um bom modelo.

Treinando-se com a massa molar do PEG e viscosidade do PEG, obteve-se:

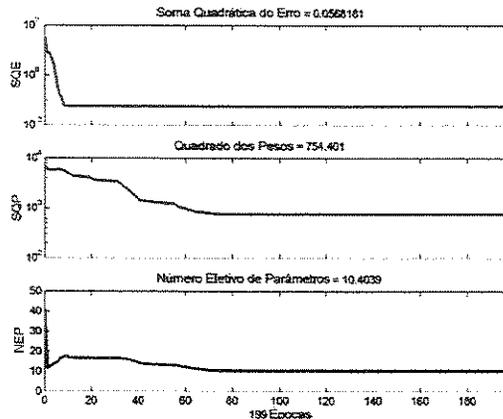


Figura 4.13- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.

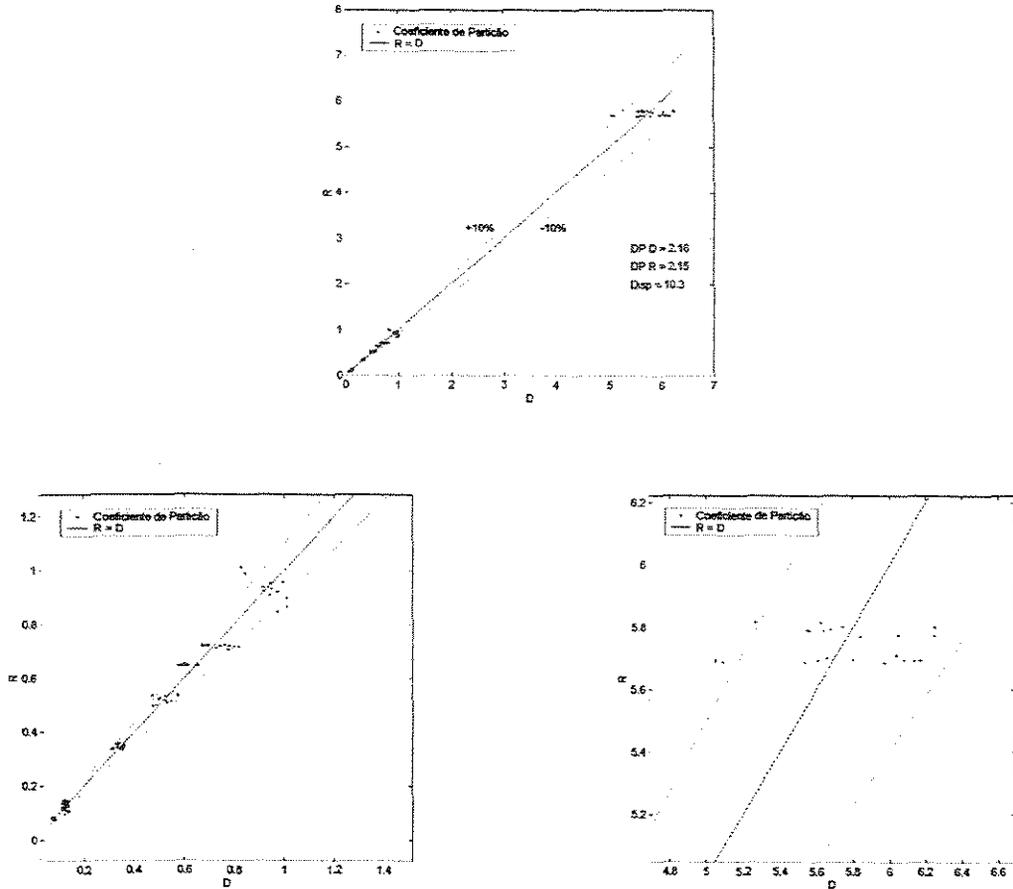


Figura 4.14- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.

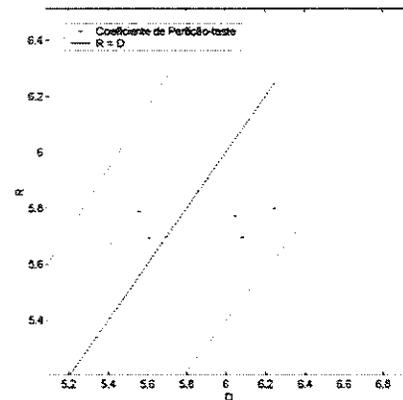
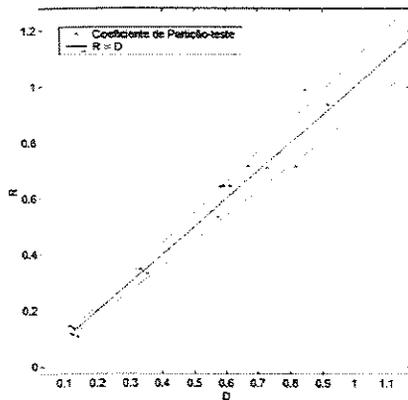
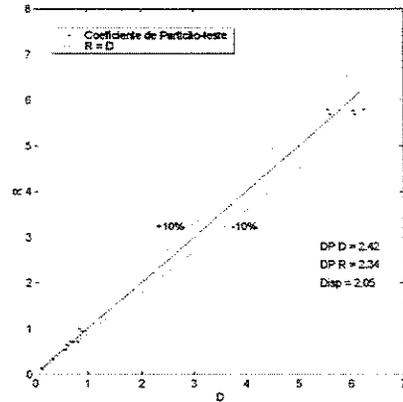


Figura 4.14- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.

Analisando-se as figuras 4.13 e 4.14 observou-se uma baixa dispersão em relação ao modelo anterior e os dados estão na faixa de $\pm 10\%$, indicando que a variável massa específica não é necessária.

Treinando-se com a massa molar do PEG e massa específica do PEG, obteve-se:

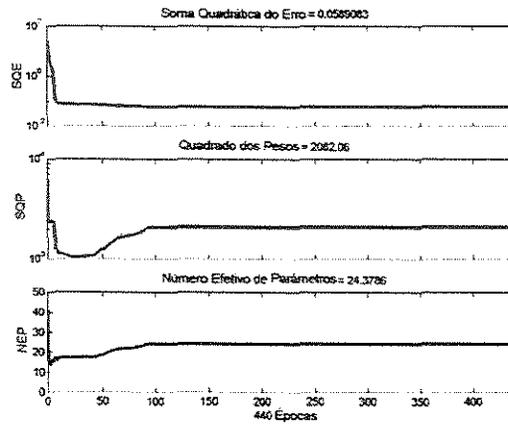


Figura 4.15- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.

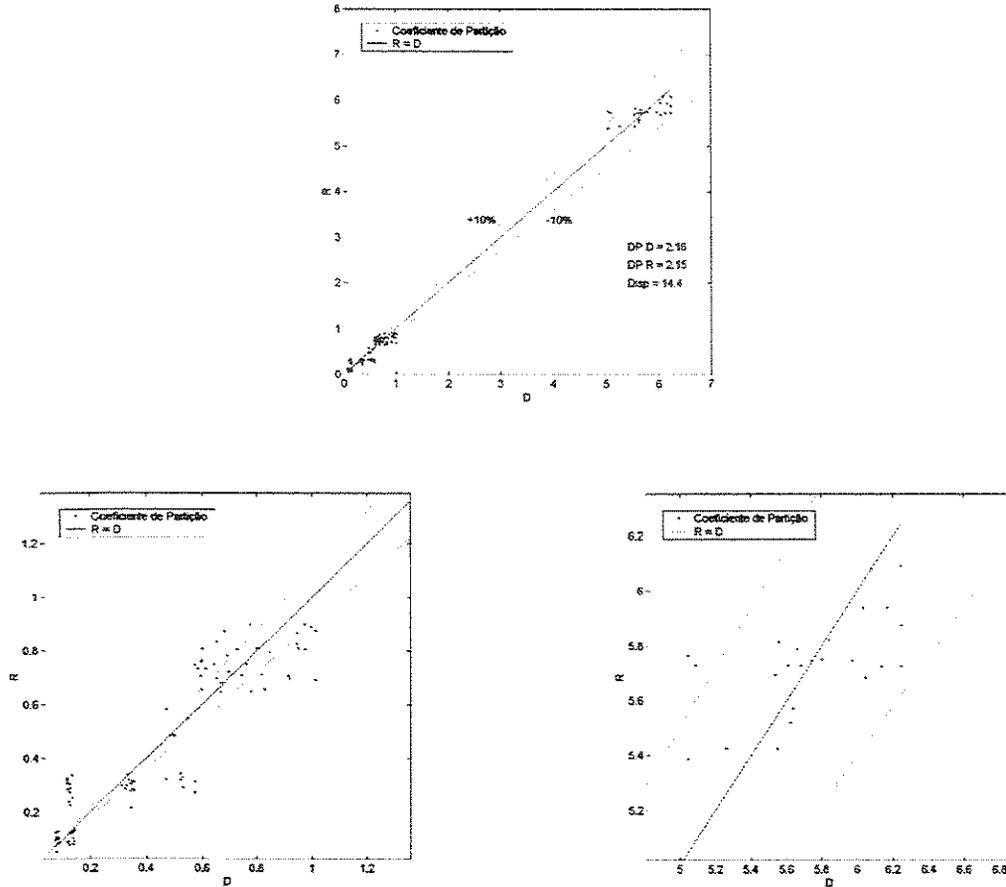


Figura 4.16- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.

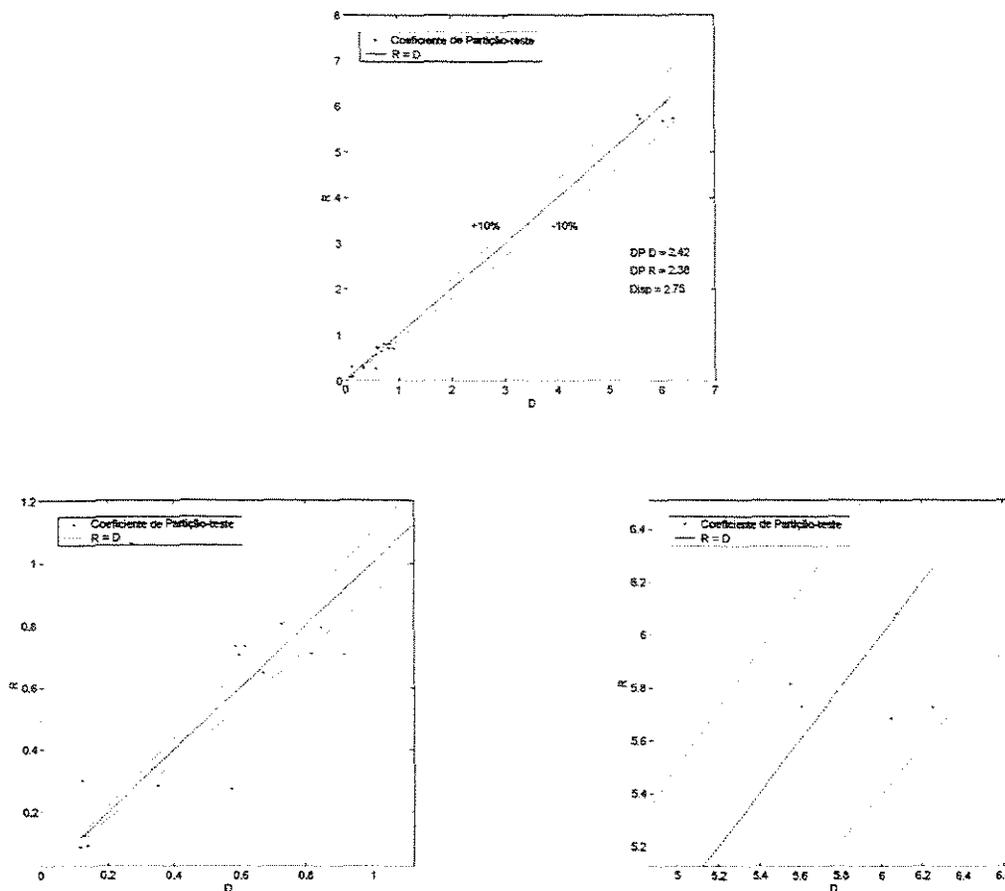


Figura 4.17- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.

Analisando-se as figuras 4.16 e 4.17 nota-se que houve um aumento na dispersão e muitos dados ficaram fora da faixa de erro de $\pm 10\%$ confirmando que a variável massa específica da fase PEG influencia pouco no modelo, e que a variável viscosidade da fase PEG é necessária.

Treinando-se com a massa molar do PEG:

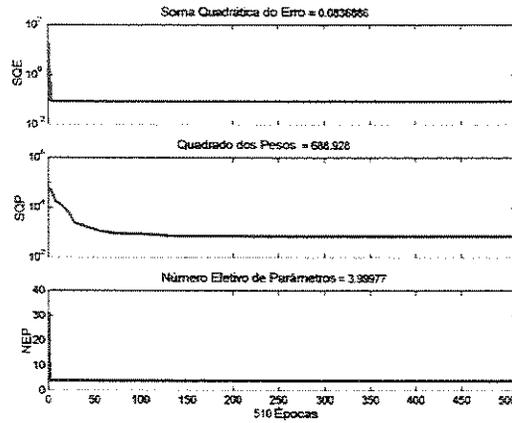


Figura 4.18- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.

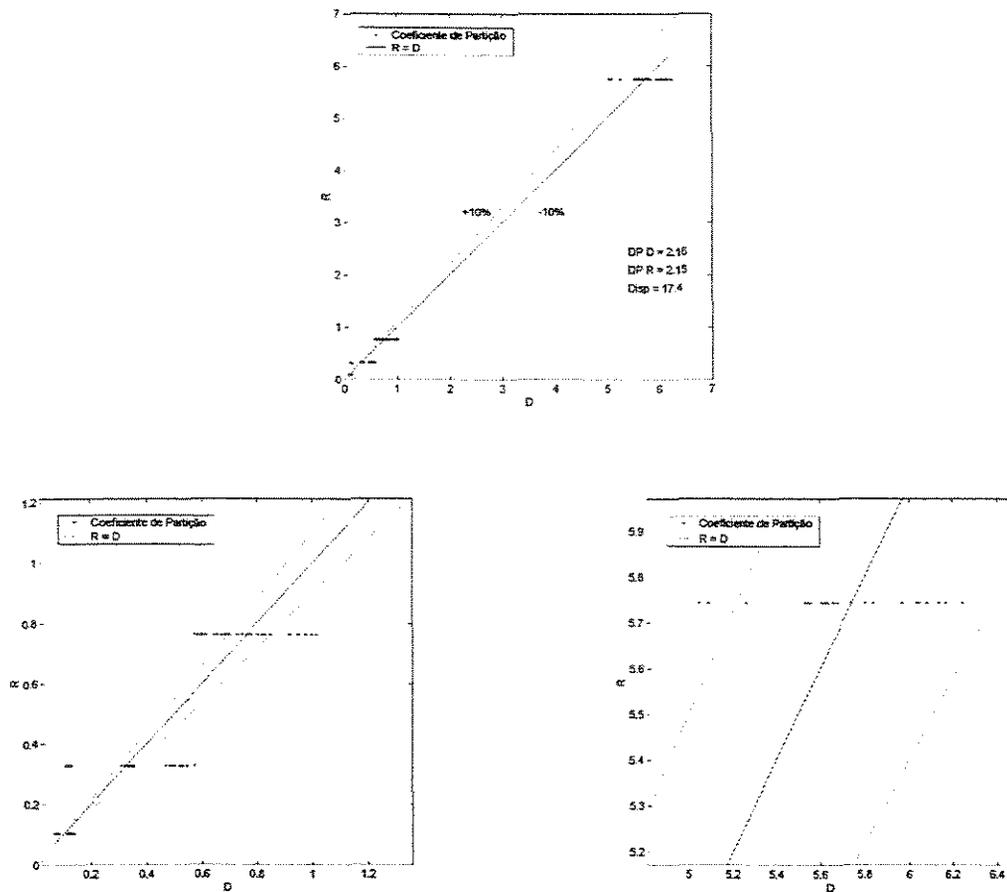


Figura 4.19- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.

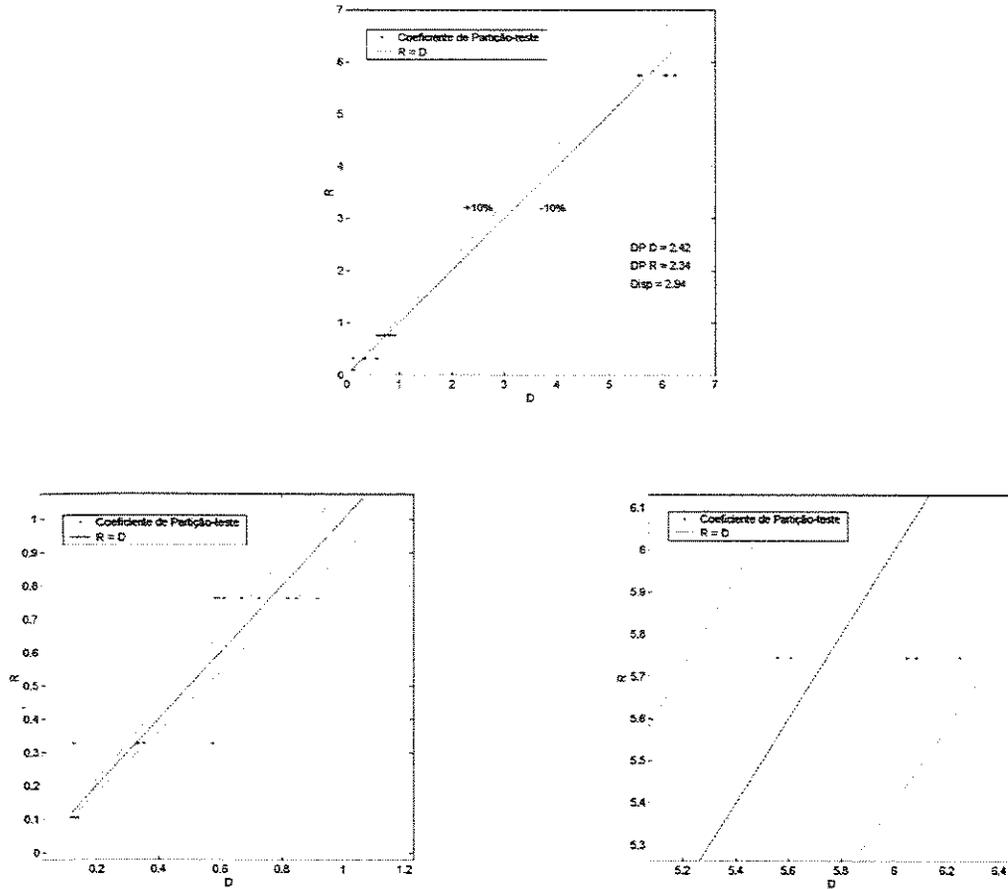


Figura 4.20- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.

Analisando-se as figuras 4.19 e 4.20 conclui-se que o modelo precisa de outras variáveis, e que somente a massa molar do PEG não é capaz de modelar o sistema.

Treinando-se a rede com a massa molar do PEG, massa específica do PEG e massa específica do sal:

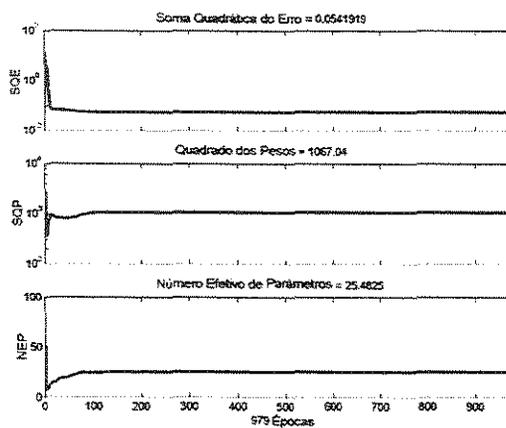


Figura 4.21- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.

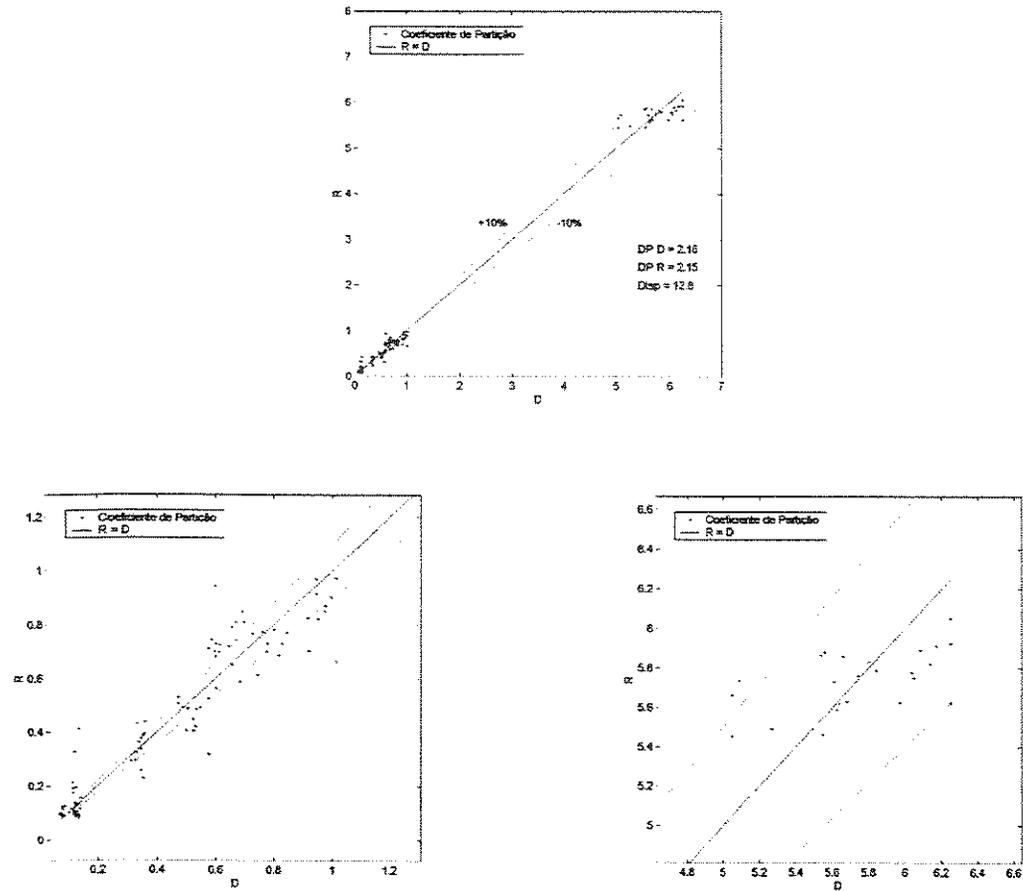


Figura 4.22- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.

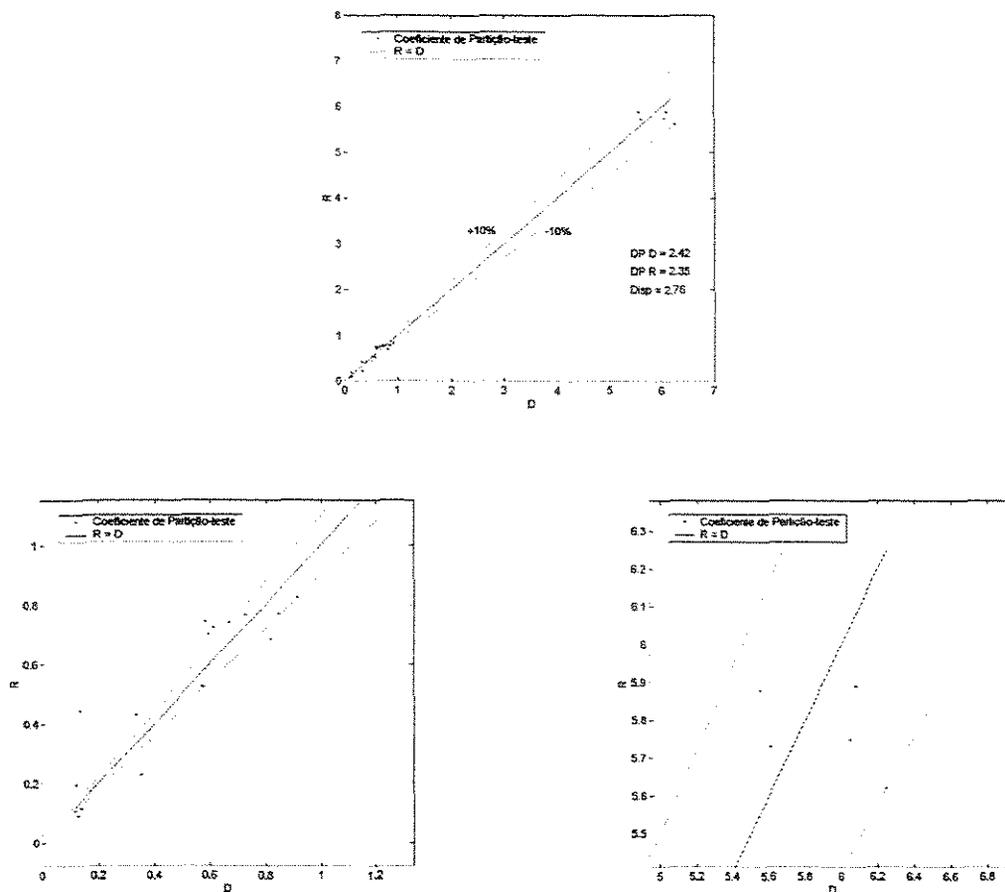


Figura 4.23- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.

Analisando-se as figuras 4.22 e 4.23 notamos que a dispersão é alta e que muitos dados estão fora da faixa de $\pm 10\%$, mostrando que as massas específicas não são variáveis adequadas para o modelo, esta resposta já era esperada devido às massas específicas variarem pouco e que a diferença entre elas é próxima de zero.

Treinando-se com a massa molar do PEG, viscosidade da fase PEG e viscosidade da fase sal:

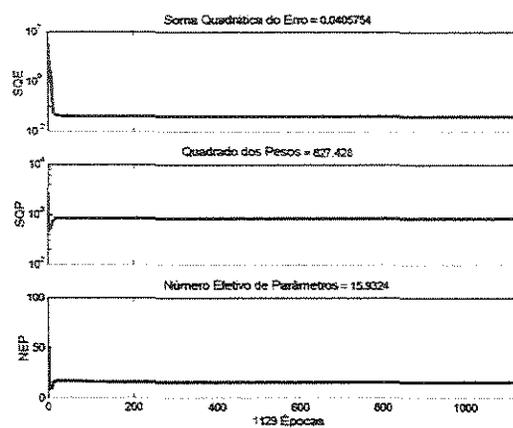


Figura 4.23- Gráfico dos parâmetros de treinamento para seleção das variáveis em operação descontínua.

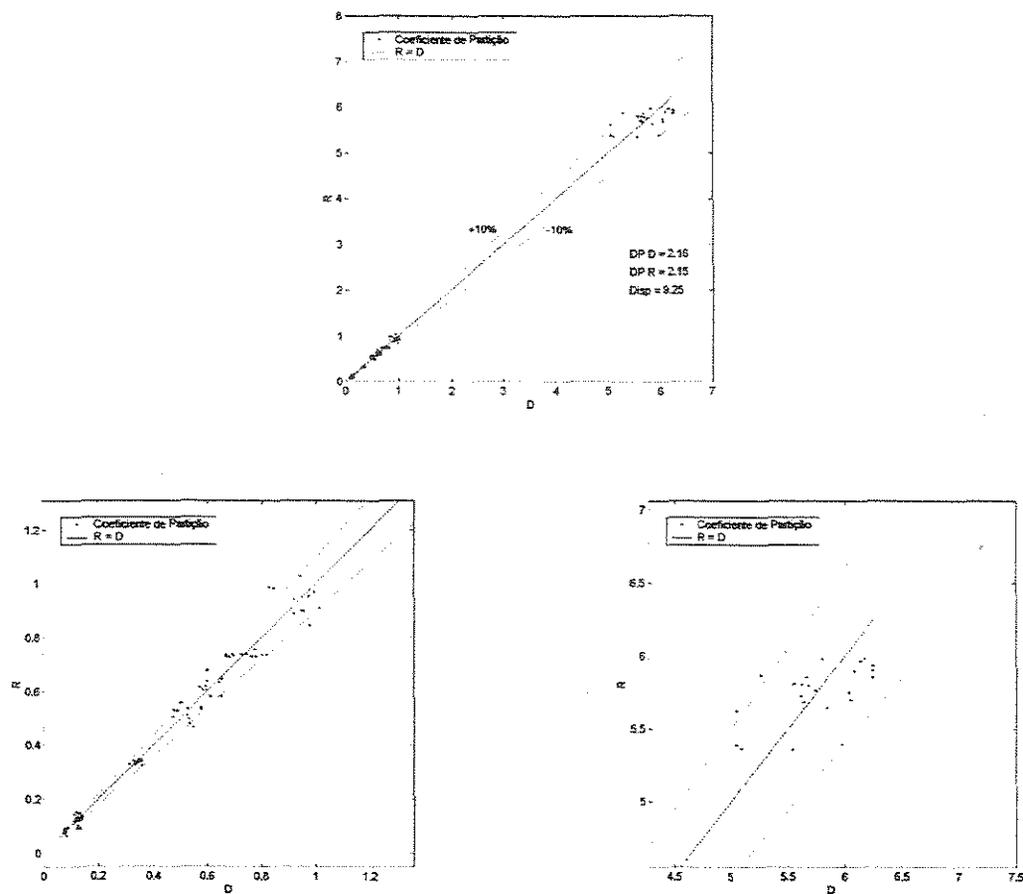


Figura 4.24- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de treinamento.

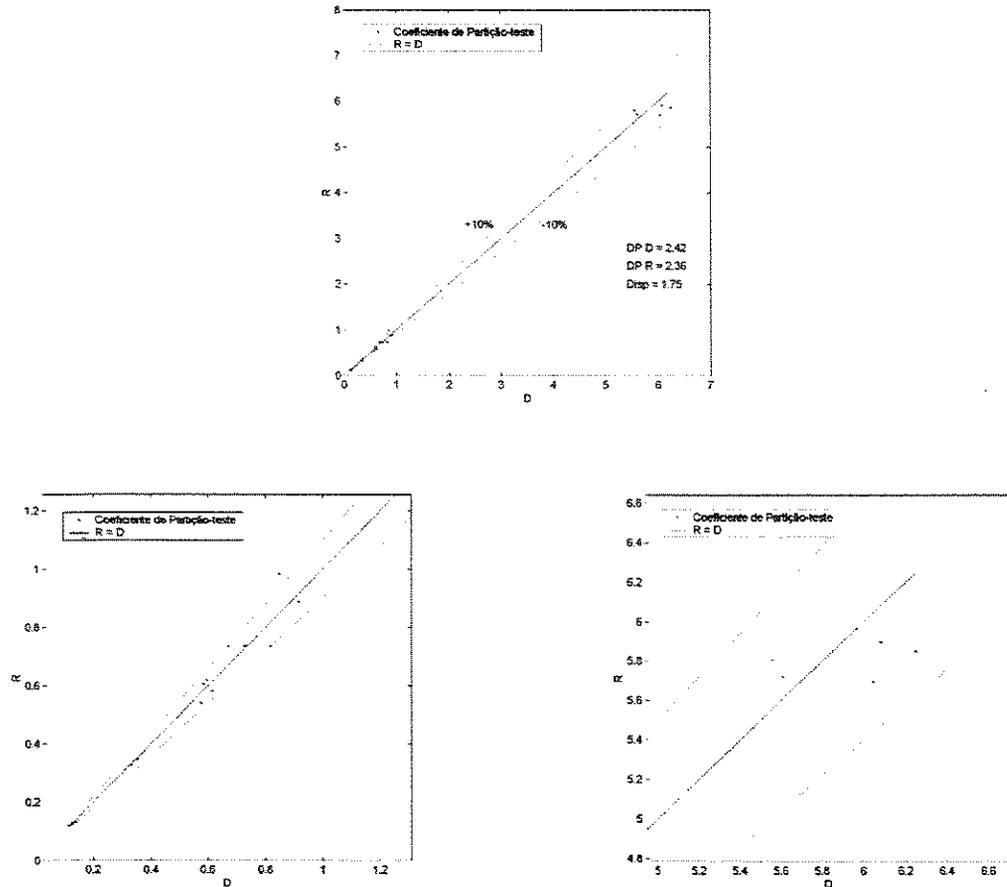


Figura 4.25- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o coeficiente de partição do conjunto de teste.

Analisando-se as figuras 4.24 e 4.25 nota-se que a dispersão é baixa e que a maioria dos dados encontra-se na faixa de $\pm 10\%$ indicando que estas variáveis são adequadas para o modelo.

Analisando-se todos os treinamentos nota-se que existe mais de um modelo capaz de representar o sistema, optou-se pelo modelo com a massa molar do PEG e as viscosidades das fases devido ter um maior significado físico, por serem estes parâmetros os que mais variam e que estas variáveis demonstram as propriedades de ambas fases.

4.8.1.4 Treinamento do Modelo

Com o objetivo de avaliar o desempenho dos algoritmos de treinamento, treinou-se a rede com três algoritmos: Levenberg-Marquardt, Parada Antecipada e Regularização Bayesiana, com dados com variação na entrada e sem variação na entrada, variando-se o número de neurônios da camada intermediária para encontrar o melhor modelo.

4.8.1.4.1 Treinamento com dados com variação na entrada

4.8.1.4.1.1 Algoritmo Levenberg-Marquardt

Tabela 4.5 - Treinamento com Algoritmo Levenberg-Marquardt

Levenberg-Marquardt			Conj. Treinamento			Conj. Teste		
Neurônios	Iterações	Des*10 ⁻⁴	Disp	DP R	DP D	Disp	DP R	DP D
2	*	3,39	8,51	2,1	2,11	2,08	2,36	2,42
4	*	2,92	6,39	2,1	2,11	1,83	2,35	2,42
6	*	1,4	6,1	2,11	2,11	2,01	2,38	2,42
8	*	1,37	4,69	2,1	2,11	8,58	2,57	2,42
10	230	1	4,15	2,1	2,11	2,69	2,25	2,42

* não alcançou o desempenho desejado até o número máximo de iterações (2000).

Disp – Dispersão dos dados

DP R – Desvio padrão dos dados calculados pela rede

DP R – Desvio padrão dos dados experimentais

Des – Desempenho da rede

Desempenho desejado = 1.10^{-4}

Através da tabela 4.5 conclui-se que a melhor arquitetura é 3-10-1, que é a rede com menor número de neurônios capaz de alcançar o desempenho desejado de 1.10^{-4} .

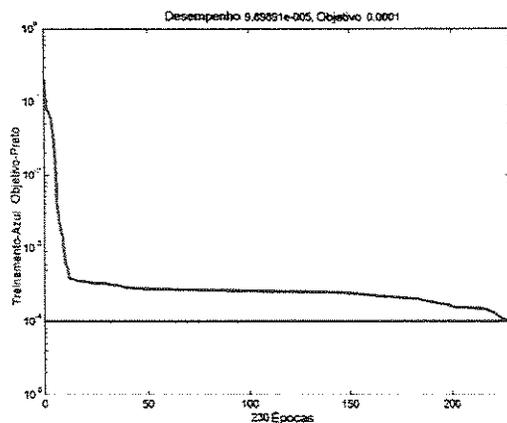


Figura 4.26- Gráfico do desempenho do Algoritmo Levenberg-Marquardt durante treinamento.

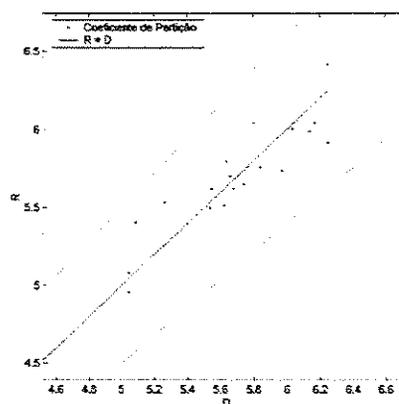
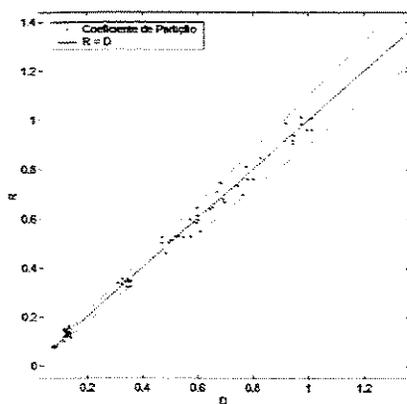
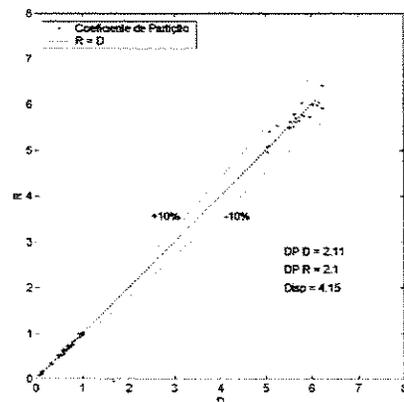


Figura 4.27- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para o treinamento da Operação Descontínua - com dados com variação na entrada.

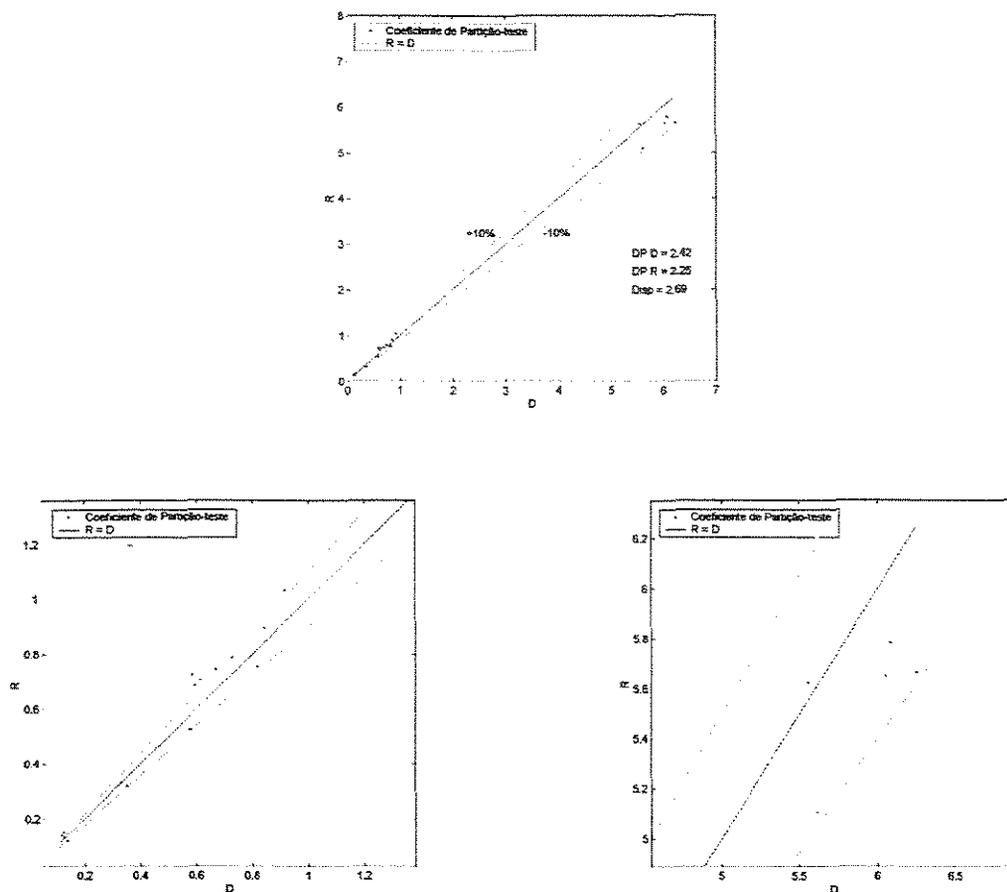


Figura 4.28- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste do modelo da Operação Descontínua - com dados com variação na entrada.

A figura 4.27 mostra que os dados de treinamento tiveram uma baixa dispersão e a maioria dos dados encontra-se na faixa de erro de $\pm 10\%$, como mostrado na ampliação das áreas do gráfico.

A figura 4.28 mostra que os dados de teste apresentaram uma baixa dispersão e que o erro ficou entre $\pm 10\%$, indicando que não houve sobreajuste nos dados.

4.8.1.4.1.2 Algoritmo Parada Antecipada:

Tabela 4.6 - Treinamento com Algoritmo Parada Antecipada.

Parada Antecipada			Conj. Treinamento			Conj. Teste		
Neurônios	Iterações	Des*10 ⁻⁴	Disp	DP R	DP D	Disp	DP R	DP D
2	31	8,3	14,7	2,08	2,1	3,53	2,31	2,42
4	24	5,8	11	2,06	2,1	3,1	2,33	2,42
6	25	3,8	6,9	2,08	2,1	1,94	2,35	2,42
8	23	3,7	6,53	2,09	2,1	1,92	2,35	2,42
10	28	3,8	6,06	2,09	2,1	1,96	2,36	2,42
12	21	3,8	7,56	2,07	2,1	2,06	2,35	2,42

Disp – Dispersão dos dados

DP R – Desvio padrão dos dados calculados pela rede

DP R – Desvio padrão dos dados experimentais

Des – Desempenho da rede

Desempenho desejado = 1.10^{-4}

Analisando-se os dados da tabela conclui-se que a melhor arquitetura é 3-8-1, que é a menor rede que apresenta uma baixa dispersão (Disp) e o desempenho é próximo ao desejado (Des).

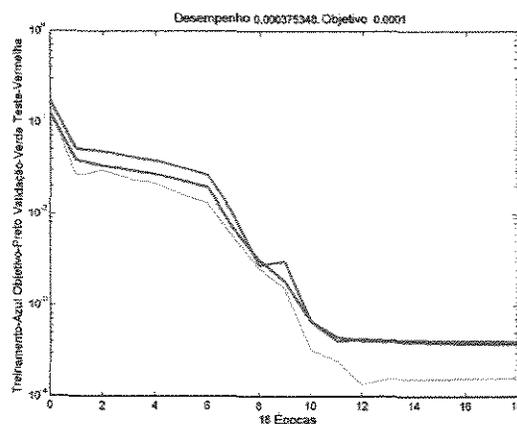


Figura 4.29 - Gráfico do desempenho do Algoritmo Parada Antecipada durante treinamento.

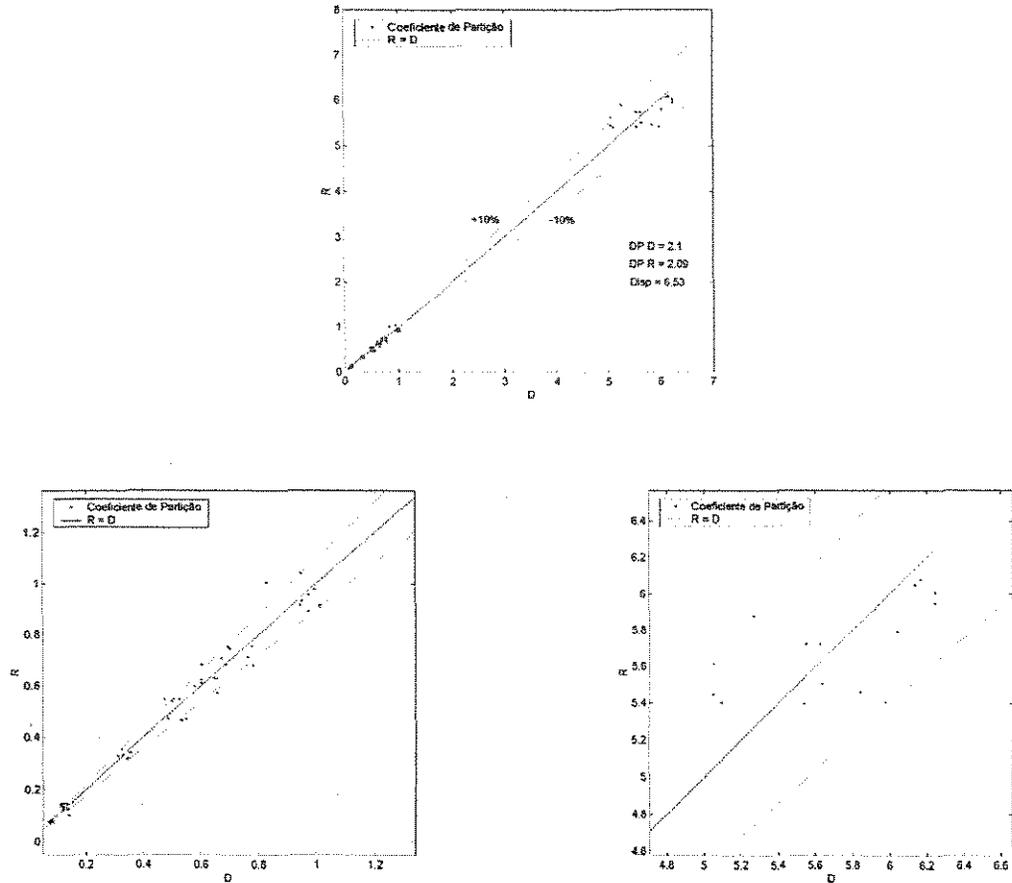


Figura 4.30- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Descontínua - com dados com variação na entrada.

Analisando-se a figura 4.30 observa-se que a dispersão é baixa e que os dados encontram-se na faixa de erro de $\pm 10\%$, mostrando que o treinamento foi satisfatório.

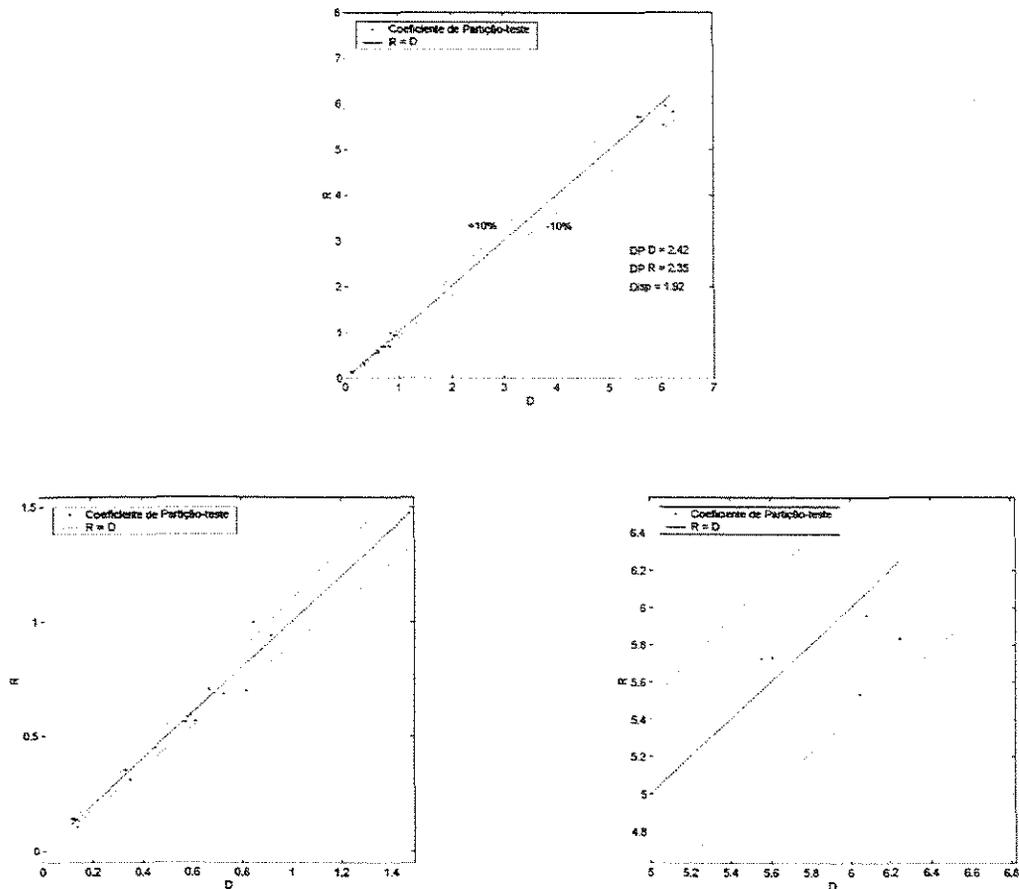


Figura 4.31 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Descontínua - com dados com variação na entrada.

Observando-se a figura 4.31 nota-se que a dispersão é baixa e com maior parte dos dados na faixa de erro de $\pm 10\%$, concluindo-se que não houve sobreajuste dos dados já que este algoritmo evita o sobreajuste, a análise visual dos gráficos prova isto devido à baixa dispersão do conjunto de teste.

4.8.1.4.1.3 Algoritmo Regularização Bayesiana:

Tabela 4.7 - Treinamento com Algoritmo Regularização Bayesiana.

Regularização Bayesiana					Conj. Treinamento			Conj. Teste		
Neurônios	Iterações	SQE	SQP	NEP	Disp	DP R	DP D	Disp	DP R	DP D
2	45	0,040	3307	9,39	9,82	2,15	2,16	1,89	2,96	2,42
4	141	0,040	1057	15,50	9,15	2,15	2,16	1,9	2,36	2,42
6	285	0,040	1040	17,40	8,86	2,15	2,16	1,78	2,36	2,42
8	661	0,037	1293	19,03	8,72	2,15	2,16	1,8	2,36	2,42
10	224	0,037	1292	19,00	8,72	2,15	2,16	1,8	2,36	2,42

SQE – Soma quadrática do erro

SQP – Soma quadrática dos pesos

NEP – Número efetivo de parâmetros

Disp – Dispersão dos dados

DP R – Desvio padrão dos dados calculados pela rede

DP R – Desvio padrão dos dados experimentais

Através da análise da tabela 4.7 conclui-se que a melhor arquitetura é 3-8-1, devido os parâmetros de treinamento ficarem constantes (SQE, SQP, NEP), indicando que este é o número necessário de neurônios.

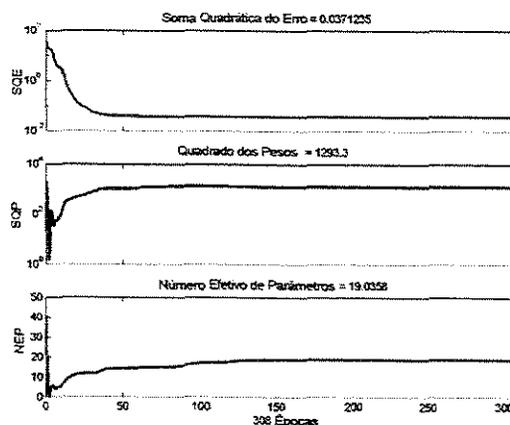


Figura 4.31- Gráfico do desempenho do Algoritmo Regularização Bayesiana durante treinamento.

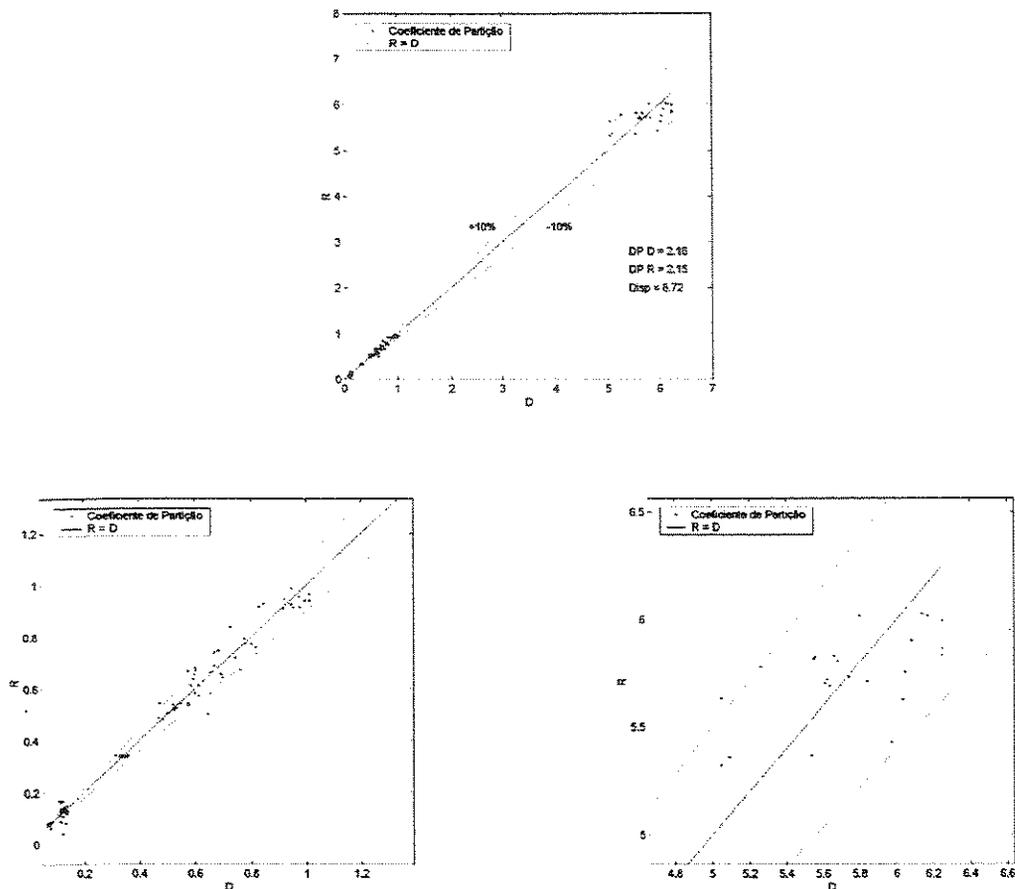


Figura 4.32 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Descontínua - com dados com variação na entrada.

Analisando-se a figura 4.32 nota-se que a dispersão é baixa e a maioria dos dados encontra-se dentro da faixa de erro de $\pm 10\%$, evidenciando que o treinamento foi satisfatório.

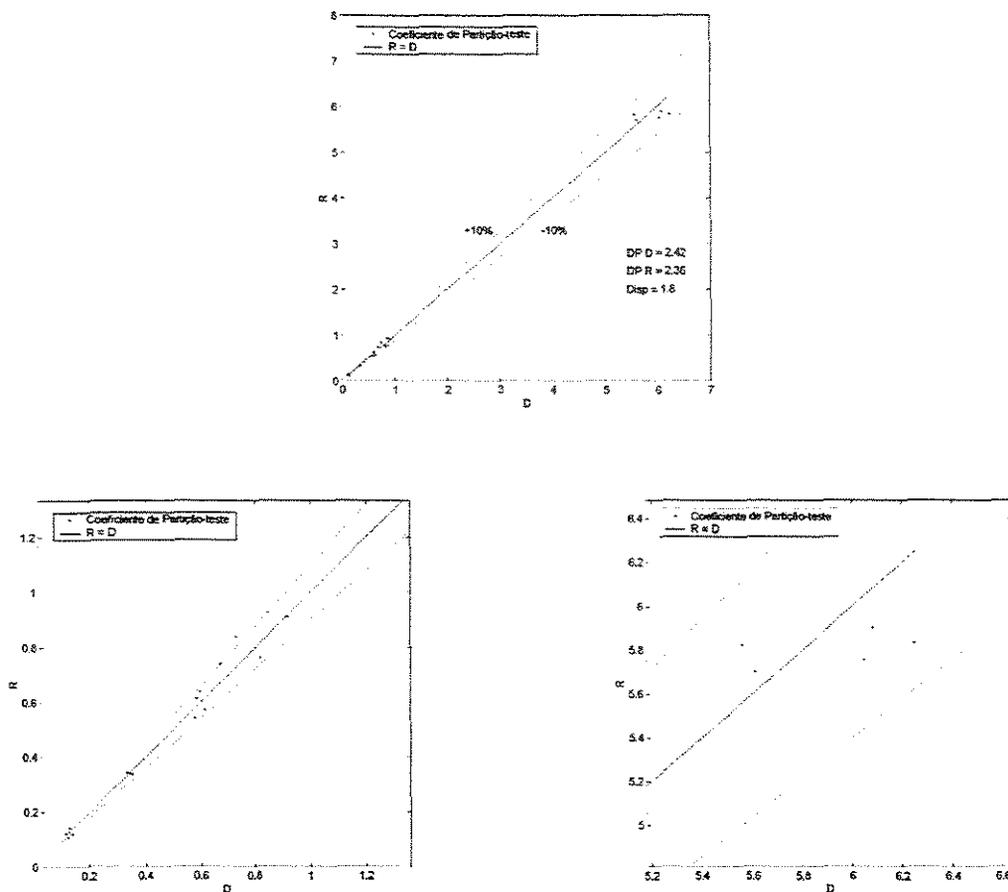


Figura 4.33 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Descontínua - com dados com variação na entrada.

Analisando-se a figura 4.33 nota-se que o conjunto de teste apresentou uma baixa dispersão e com a maioria dos dados dentro da faixa de erro $\pm 10\%$. Assim este é o algoritmo que apresenta a menor dispersão do conjunto de teste, sendo considerado o melhor algoritmo de treinamento usado. Esta vantagem é devida a este algoritmo não separar os conjuntos de treinamento e validação.

4.8.1.4.2 Treinamento com dados sem variação na entrada

4.8.1.4.2.1 Algoritmo Levenberg-Marquardt:

Tabela 4.8 - Treinamento com Algoritmo Levenberg-Marquardt

Levenberg-Marquardt			Conj. Treinamento			Conj. Teste		
Neurônios	Iterações	Des*10 ⁻⁴	Disp	DP R	DP D	Disp	DP R	DP D
2	*	3,4	8,22	2,06	2,07	1,9	2,29	2,26
4	máx μ	3,2	6,52	2,06	2,07	1,71	2,29	2,26
6	máx μ	3,2	6,52	2,06	2,07	1,71	2,29	2,26
8	máx μ	3,2	6,52	2,06	2,07	1,71	2,29	2,26
10	máx μ	3,2	6,52	2,06	2,07	1,71	2,29	2,26

* não alcançou o desempenho desejado até o número máximo de iterações (2000).

Disp – Dispersão dos dados

DP R – Desvio padrão dos dados calculados pela rede

DP R – Desvio padrão dos dados experimentais

Des – Desempenho da rede

Desempenho desejado = 1.10^{-4}

Analisando-se a tabela 4.8 nota-se que o algoritmo não alcança o desempenho desejado, mas apresenta baixas dispersões nos conjuntos de treinamento e teste.

Este comportamento é resultado de não haver dados redundantes na entrada, diminuindo-se a dimensão do vetor de entrada e eliminando-se os dados que contribuem pouco para o conjunto de dados.

Quando a rede converge para máximo μ indica que o passo do gradiente é muito pequeno e a função objetivo (rede) chegou a um mínimo.

Assim usou-se a menor rede capaz de chegar mais próximo ao desempenho desejado, ou seja 3-4-1.

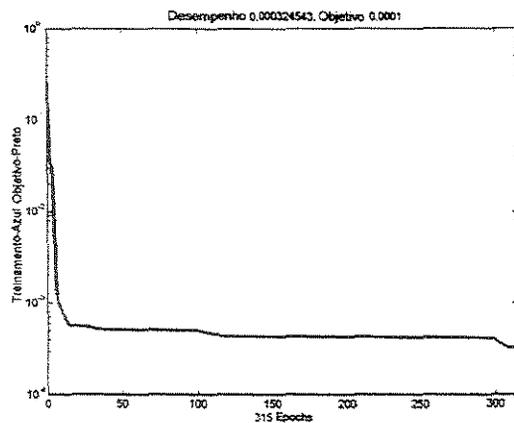


Figura 4.34- Gráfico do desempenho do Algoritmo Levenberg-Marquardt durante treinamento.

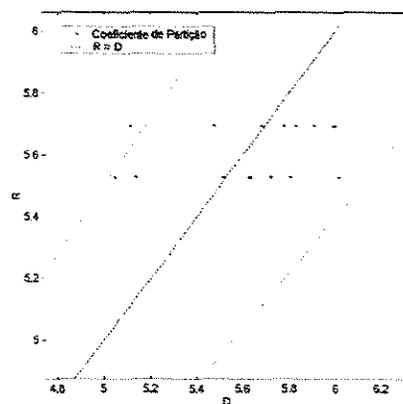
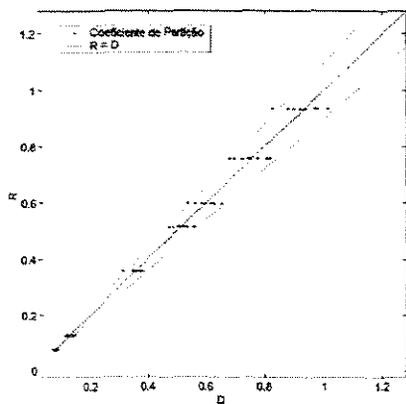
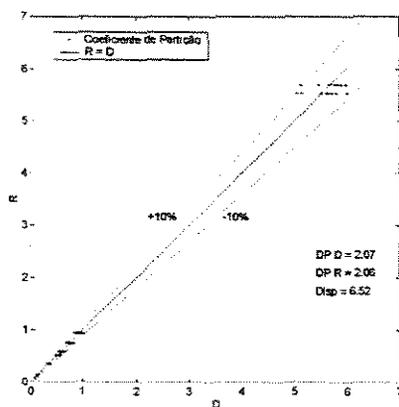


Figura 4.35 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Descontínua - com dados sem variação na entrada.

Analisando-se a figura 4.35 observou-se que a dispersão é baixa, mas superior à rede com dados com variação na entrada, porém existem muito poucos dados fora da faixa de erro de $\pm 10\%$, levando-se a concluir que com os dados com variação na entrada houve um “leve” sobreajuste na rede.

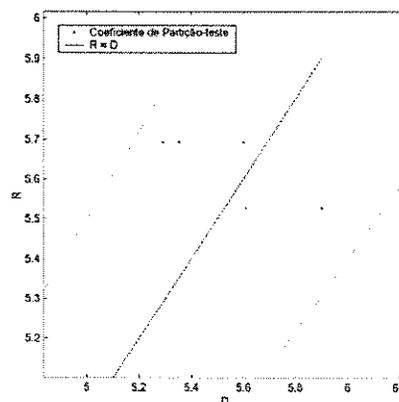
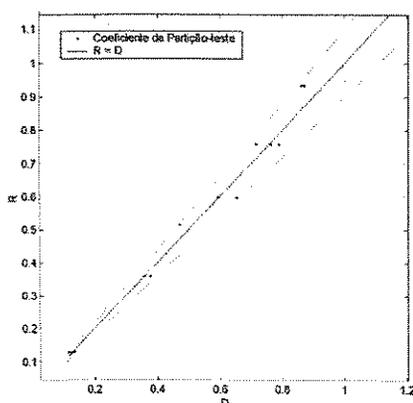
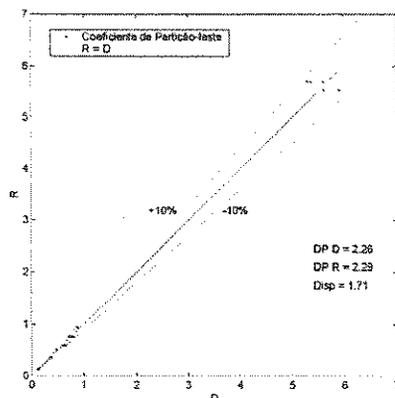


Figura 4.36- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Descontínua - com dados sem variação na entrada.

Analisando-se figura 4.36 nota-se que os dados de teste apresentaram uma baixa dispersão, menor que com os treinamentos com dados sem variação na entrada, e que não houve dados fora da faixa de erro de $\pm 10\%$, indicando que com dados sem variação na entrada não houve sobreajuste, devido à rede ter chegado ao mínimo, concluindo-se que este modelo está apto a prever o coeficiente de partição.

4.8.1.4.2.2 Algoritmo Parada Antecipada:

Tabela 4.9 - Treinamento com Algoritmo Parada Antecipada.

Parada Antecipada			Conj. Treinamento			Conj. Teste		
Neurônios	Iterações	Des*10 ⁻⁴	Disp	DP R	DP D	Disp	DP R	DP D
2	28	7,1	13,4	2,05	2,06	2,53	2,28	2,26
4	23	3,5	6,23	2,06	2,06	1,66	2,28	2,26
6	16 máx μ	3,4	5,51	2,06	2,06	1,59	2,29	2,26
8	11 máx μ	3,4	5,51	2,06	2,06	1,59	2,29	2,26
10	11 máx μ	3,4	5,51	2,06	2,06	1,59	2,29	2,26

Disp – Dispersão dos dados

DP R – Desvio padrão dos dados calculados pela rede

DP R – Desvio padrão dos dados experimentais

Des – Desempenho da rede

Desempenho desejado = 1.10^{-4}

Analisando-se a tabela 4.9 conclui-se que a melhor arquitetura é 3-6-1, devido à rede chegar ao máximo μ , concluindo-se que o algoritmo parou antes do aumento do erro do conjunto de validação. Deste modo o algoritmo comporta-se como um Levenberg-Marquardt puro. Com isso, o algoritmo não é adequado para trabalhar com os dados da operação descontínua sem variação na entrada.

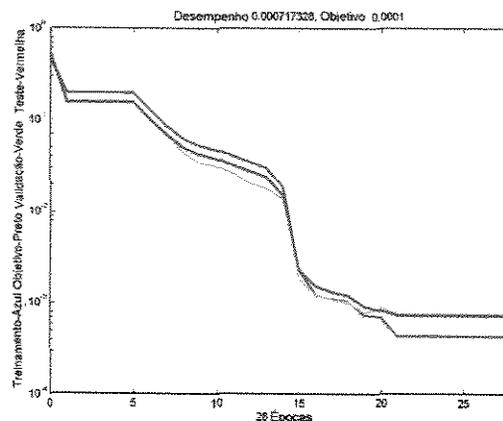


Figura 4.37- Gráfico do desempenho do Algoritmo Parada Antecipada durante treinamento.

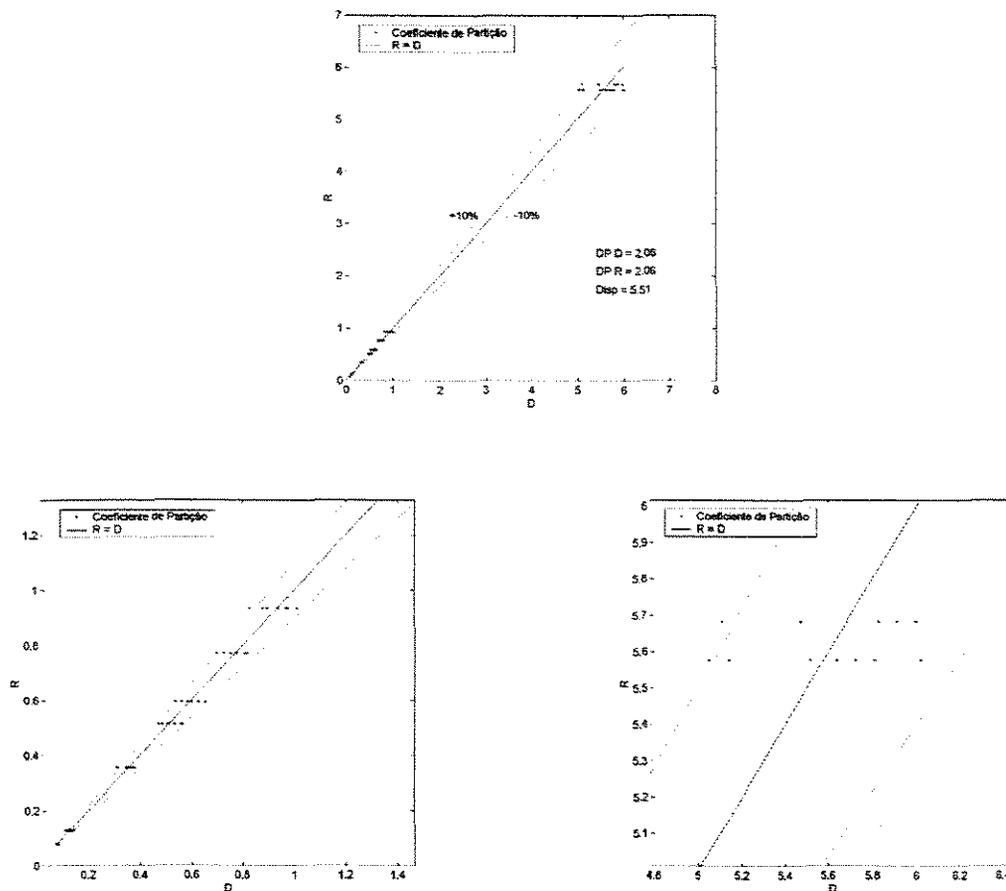


Figura 4.38- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Descontínua - com dados sem variação na entrada.

A figura 4.38 mostra que a dispersão é menor em relação ao mesmo algoritmo com dados com variação na entrada, mesmo parando antes do aumento do erro do conjunto de validação, devido não haver dados redundantes na entrada.

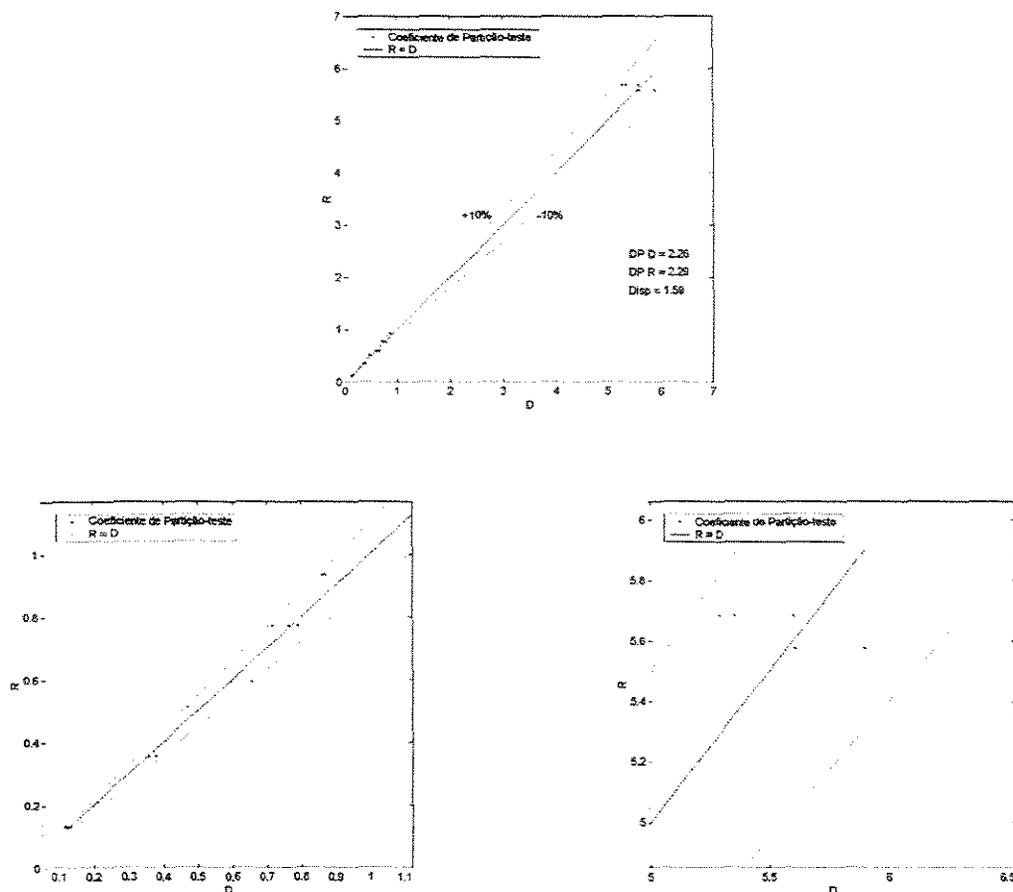


Figura 4.39 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Descontínua - com dados sem variação na entrada.

Observando-se a figura 4.39 nota-se que a dispersão também é baixa e com maior parte dos dados na faixa de erro de $\pm 10\%$, demonstrando que não houve sobreajuste.

Para este conjunto de dados, o algoritmo Parada Antecipada não é necessário, pois apresenta o mesmo comportamento de um Levenberg-Marquardt.

4.8.1.4.2.3 Algoritmo Regularização Bayesiana:

Tabela 4.10- Treinamento com Algoritmo Regularização Bayesiana.

Regularização Bayesiana					Conj. Treinamento			Conj. Teste		
Neurônios	Iterações	SQE	SQP	NEP	Disp	DP R	DP D	Disp	DP R	DP D
2	103	0,042	2538	7,53	10,2	2,09	2,09	1,52	2,28	2,26
4	255	0,039	572	9,96	8,03	2,09	2,09	1,36	2,27	2,26
6	485	0,039	557	9,99	8,03	2,09	2,09	1,37	2,27	2,26
8	494	0,039	557	9,99	8,03	2,09	2,09	1,37	2,27	2,26
10	1512	0,039	557	9,99	8,03	2,09	2,09	1,37	2,27	2,26

Disp – Dispersão dos dados

DP R – Desvio padrão dos dados calculados pela rede

DP R – Desvio padrão dos dados experimentais

Des – Desempenho da rede

SQE – Soma quadrática do erro

SQP – Soma quadrática dos pesos

NEP – Número efetivo de parâmetros

Através da análise da tabela 4.10 conclui-se que a melhor arquitetura é 3-6-1, devido os parâmetros de treinamento ficarem constantes, indicando que este é o número necessário de neurônios.

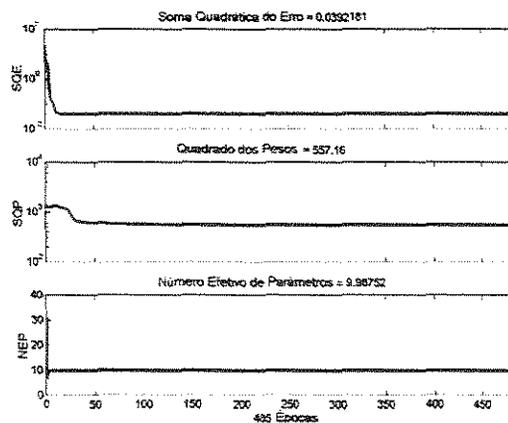


Figura 4.39- Gráfico do desempenho do Algoritmo Regularização Bayesiana durante treinamento.

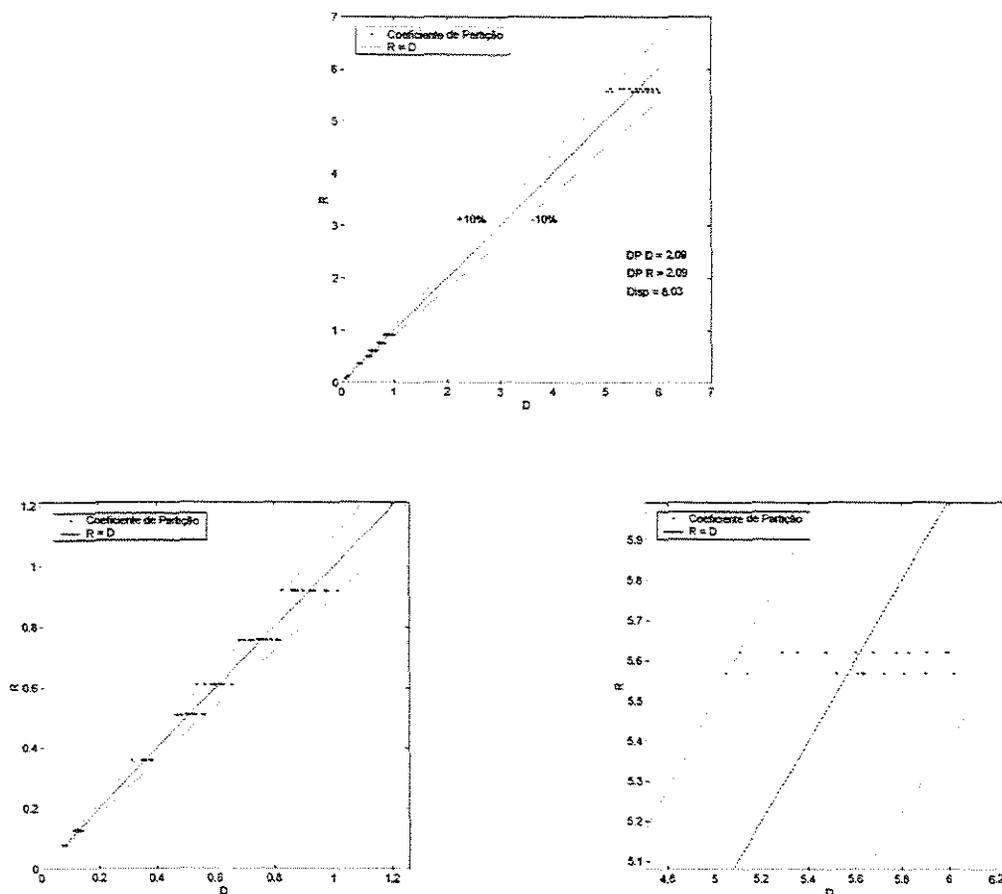


Figura 4.40 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Descontínua - com dados sem variação na entrada.

Analisando-se a figura 4.40 nota-se que a dispersão é baixa, e todos dados encontram-se dentro da faixa de erro de $\pm 10\%$, demonstrando que a rede convergiu para um mínimo.

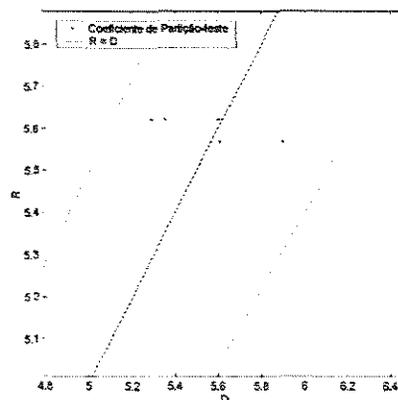
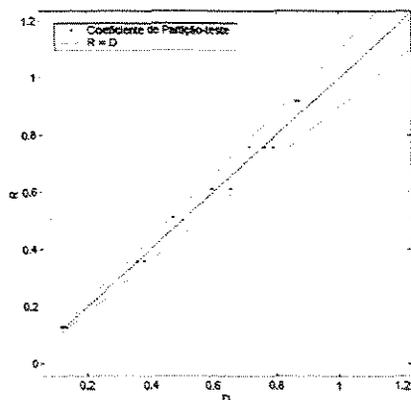
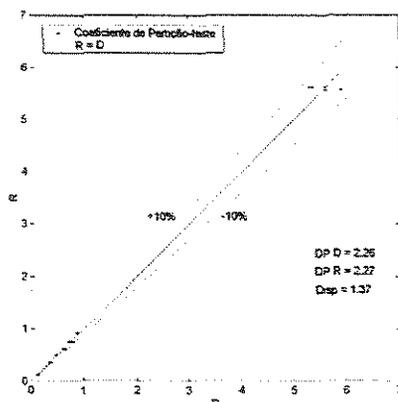


Figura 4.41 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Descontínua - com dados sem variação na entrada.

Analisando-se a figura 4.41 o conjunto de teste apresentou uma baixa dispersão e com todos os dados dentro da faixa de erro $\pm 10\%$, demonstrando que não houve sobreajuste, concluindo-se que este é o melhor algoritmo usado para treinamento com dados sem variação na entrada.

4.8.2 Operação Contínua

4.8.2.1 Seleção das Variáveis

Para verificar-se a necessidade das variáveis as redes foram treinadas com todas variáveis e depois retirando-as do modelo para verificar sua importância. O critério para exclusão das variáveis foi: se a dispersão não aumentar (ou aumentar muito pouco) com a exclusão da variável, esta não é necessária para o modelo podendo ser descartada.

Para verificação das Variáveis Operação Contínua usou-se o algoritmo Regularização Bayesiana com excesso de neurônios, e funções de ativação sigmoideal.

4.8.2.2 Variáveis

Entrada: Coeficiente de Partição, Vazão Total das Fases e Frequência.

Saída: Porcentagem de Extração e Fração de Retenção da Fase Dispersa.

Para testar-se as simplificações feitas para os dados calculados, treinou-se a rede somente com os dados calculados e depois apresentou-se os dados experimentais, se os dados experimentais ajustarem-se ao modelo significa que as simplificações são válidas.

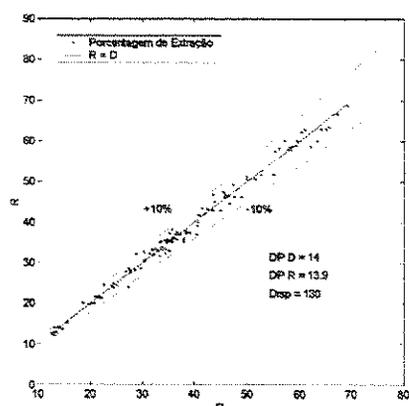
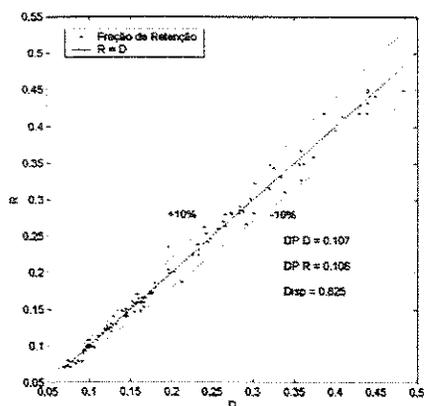
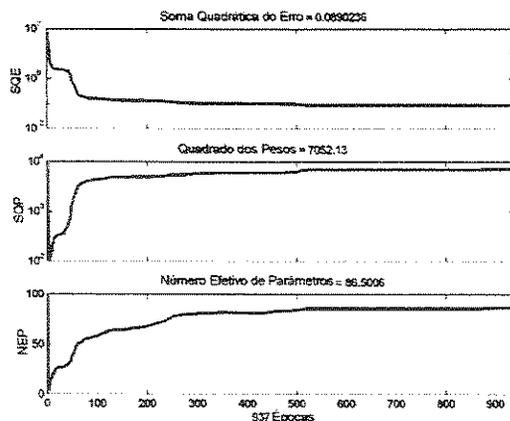


Figura 4.42- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua com dados calculados.

Através da figura 4.42 nota-se que os dados calculados tiveram um alto ajuste para o conjunto de treinamento, após verificou-se o ajuste dos dados experimentais.

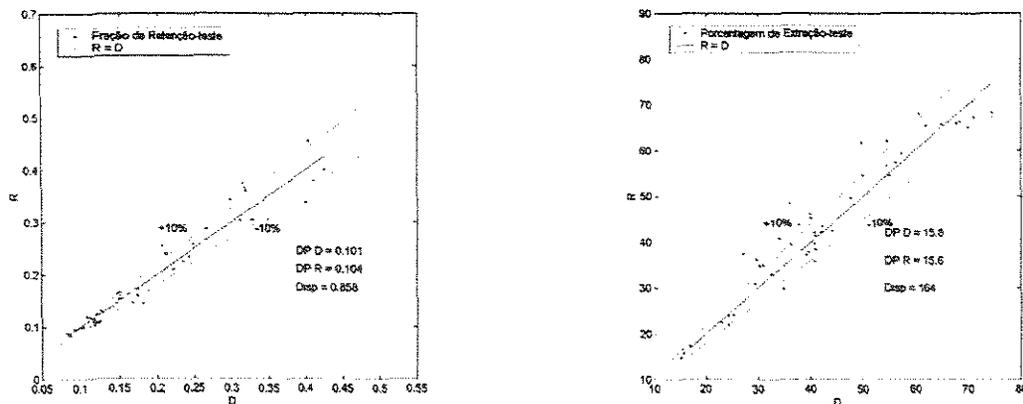


Figura 4.43- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua com dados experimentais.

Analisando-se os gráficos do conjunto de teste com os dados experimentais nota-se que existe alguns desvios provocados pelas simplificações, mas a maioria dos dados localiza-se na faixa de $\pm 10\%$ indicando que os dados calculados são válidos.

4.8.2.3 Treinamento para Seleção das Variáveis

Para verificar-se as variáveis treinou-se a rede inicialmente com todas as variáveis de entrada, ou seja Coeficiente de Partição, Vazão Total das Fases e Frequência, obtendo-se:

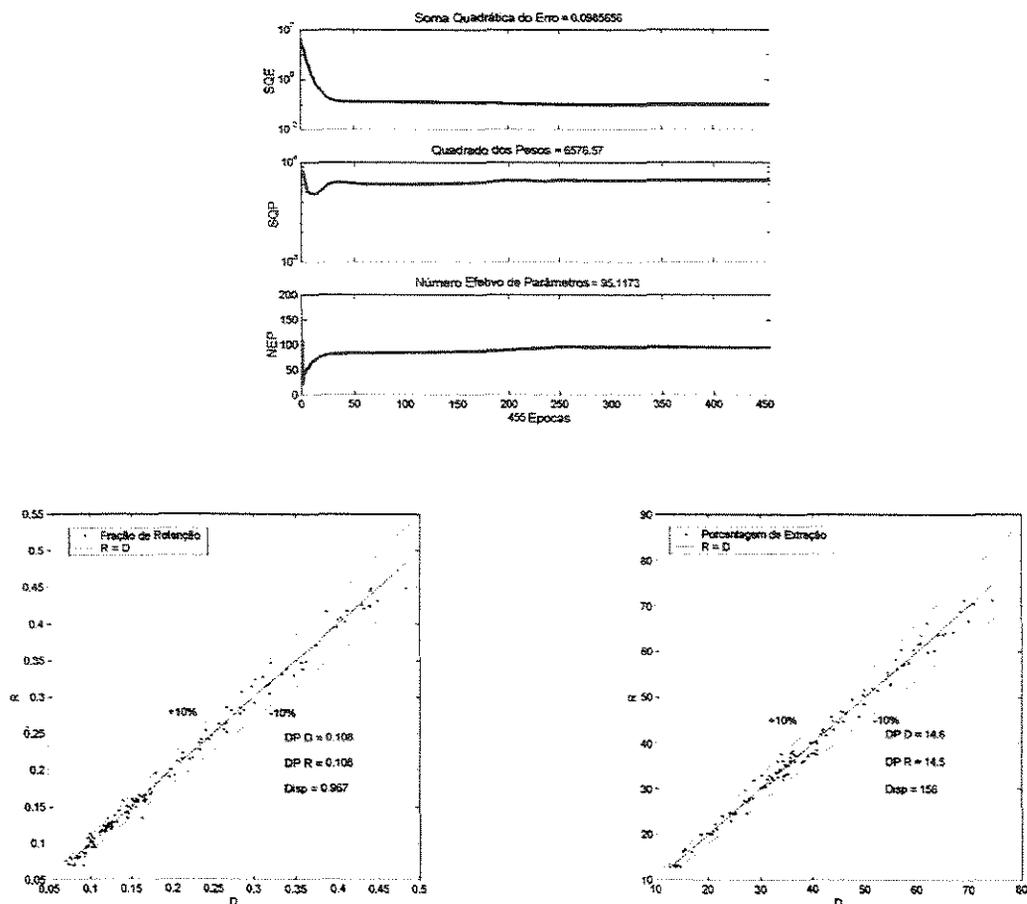


Figura 4.42- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua com Coeficiente de Partição, Vazão Total das Fases e Frequência.

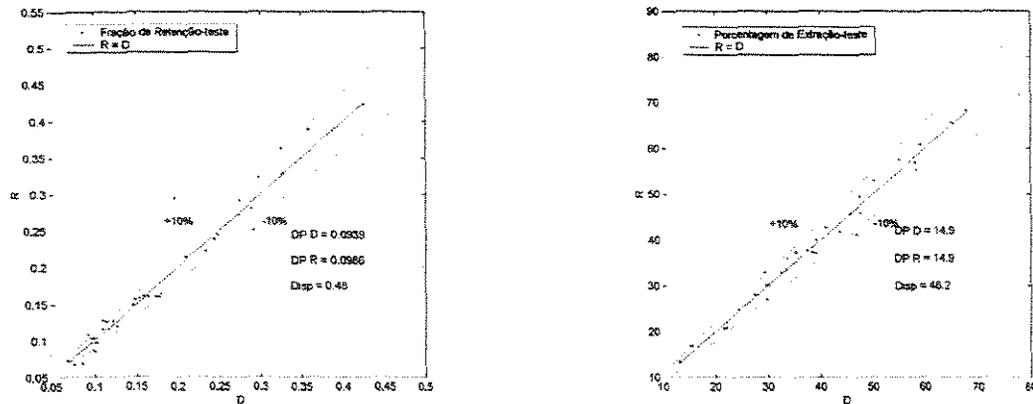


Figura 4.43- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua com Coeficiente de Partição, Vazão Total das Fases e Frequência.

Analisando-se as figuras 4.42 e 4.43 verificamos que as dispersões de ambas variáveis são baixas e que a maior parte dos dados encontra-se na faixa de erro de $\pm 10\%$, indicando que as variáveis são necessárias.

Treinando-se a rede com as variáveis Frequência e Vazão Total das fases, obteve-se:

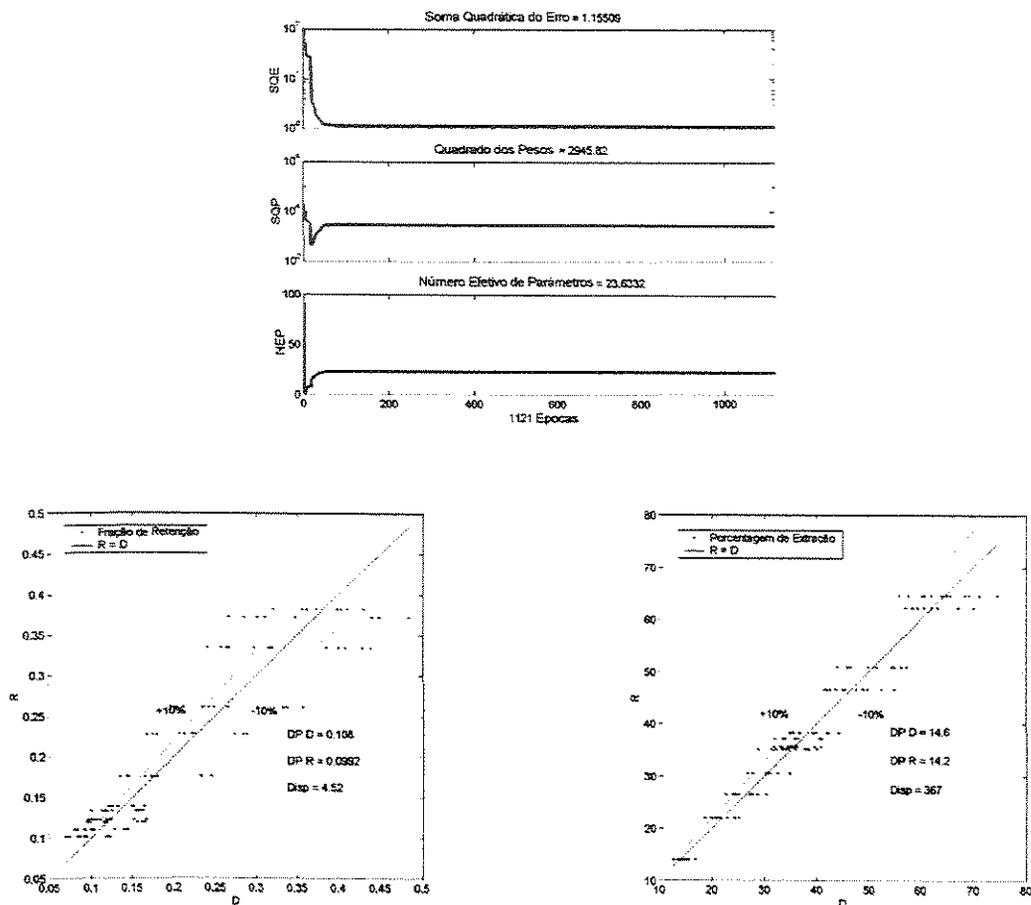


Figura 4.44- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua com Frequência e Vazão Total das fases.

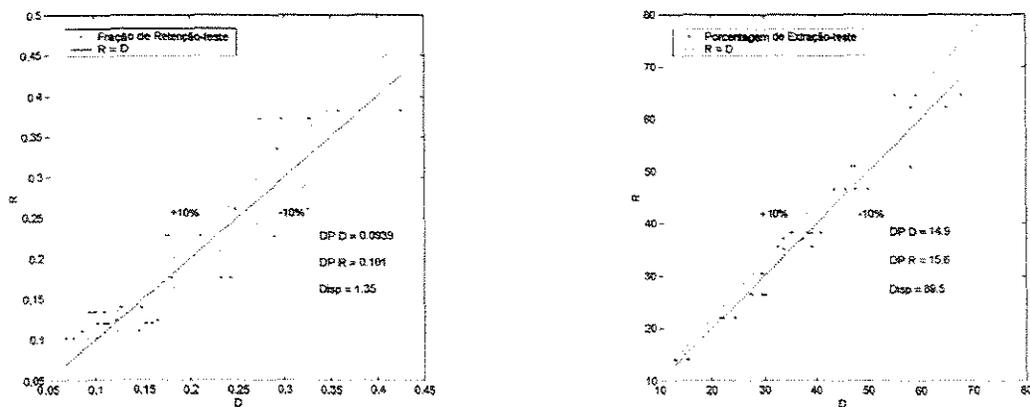


Figura 4.45 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua com Frequência e Vazão Total das fases.

Através da análise das figuras 4.44 e 4.45 nota-se que houve um aumento na dispersão dos dados e que estes se distribuíram fora da faixa de erro de $\pm 10\%$, concluindo-se que a variável coeficiente de partição é necessária ao modelo.

Treinando-se a rede com as variáveis coeficiente de partição e vazão total das fases:

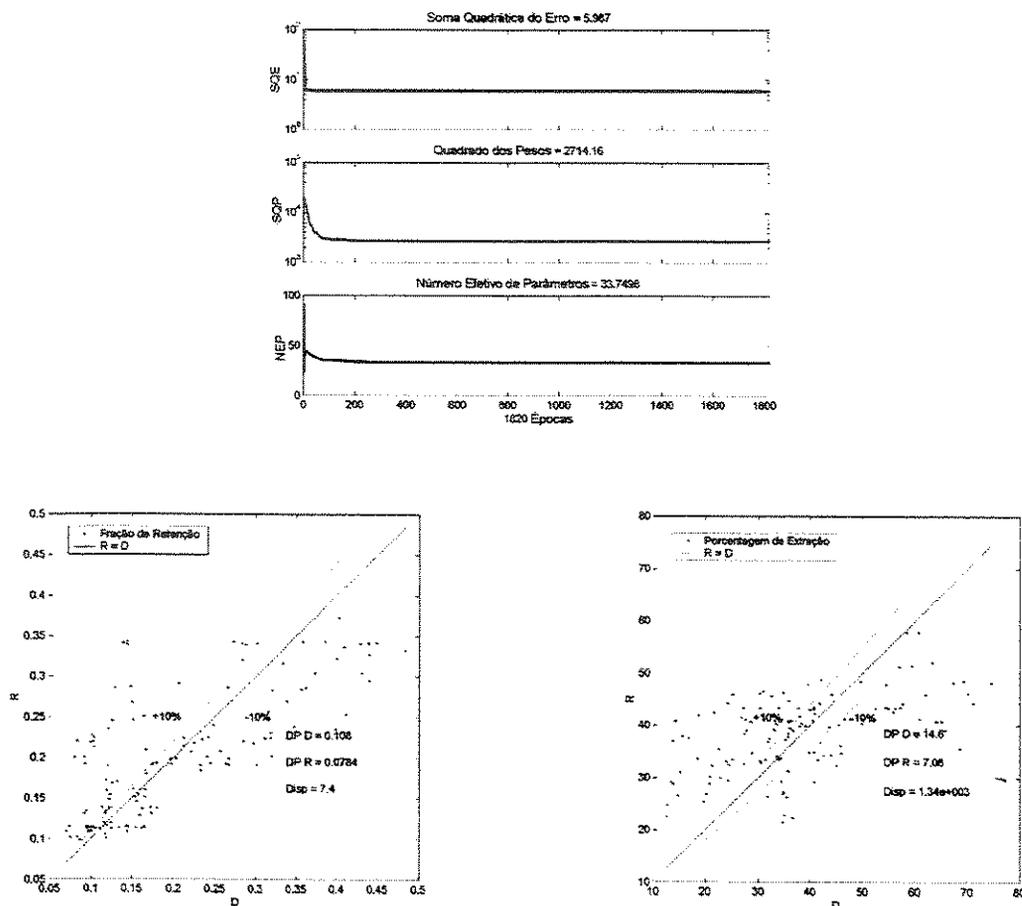


Figura 4.46 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua com Frequência e Vazão Total das fases.

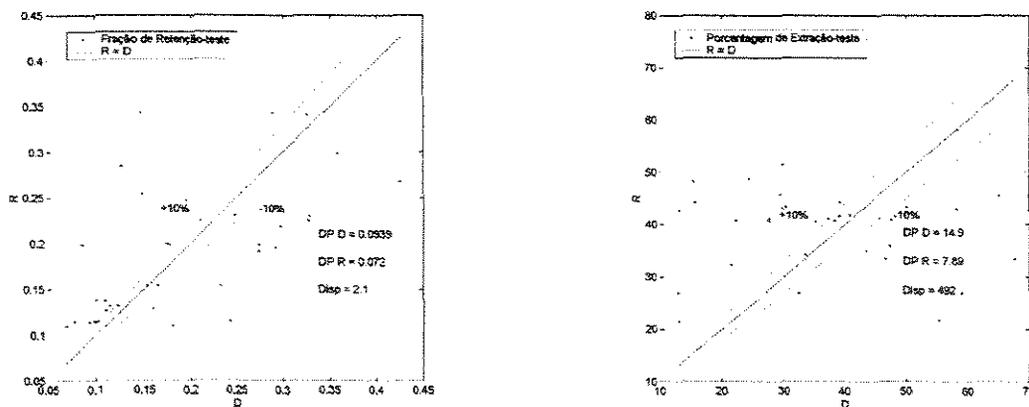


Figura 4.47- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua com Frequência e Vazão Total das fases.

A exclusão da variável frequência aumenta a dispersão, sendo que os dados distribuíram-se fora da faixa de erro de $\pm 10\%$ como indicado nas figuras 4.46 e 4.47, concluindo-se que esta variável também é importante para o modelo.

Treinando-se com as variáveis Coeficiente de Partição e Frequência:

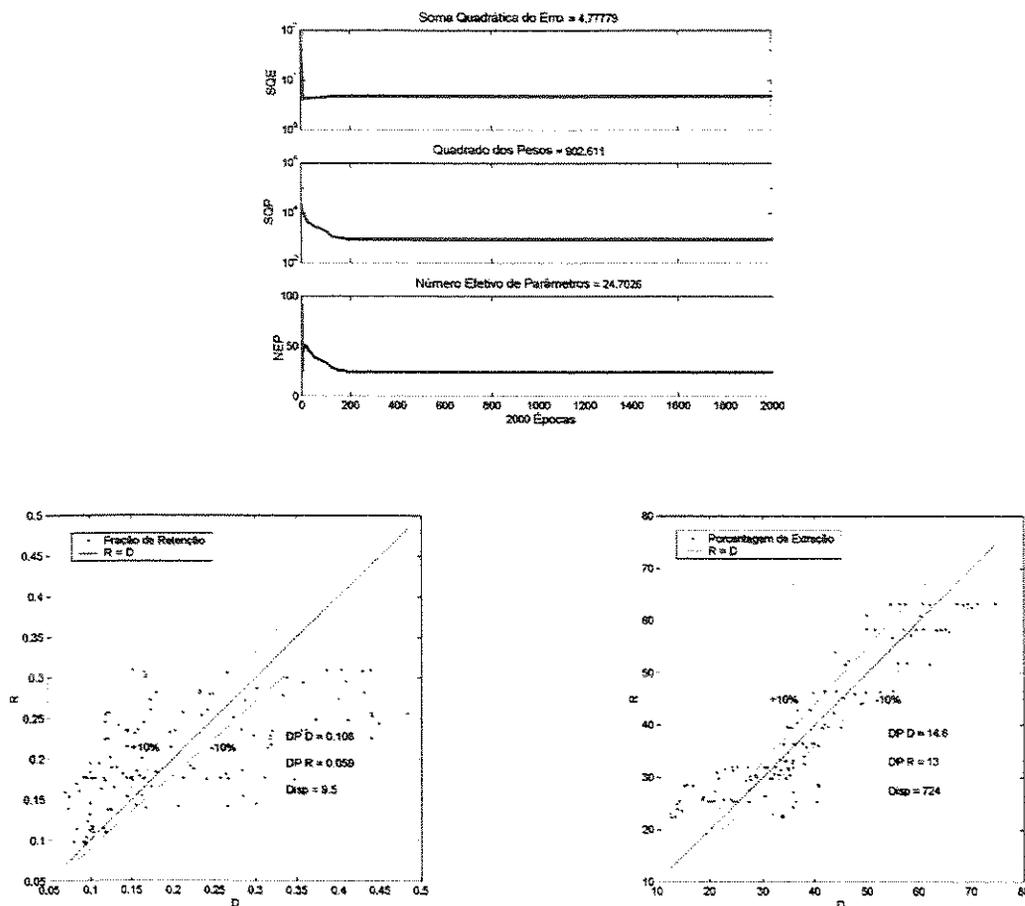


Figura 4.48- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua com Coeficiente de Partição e Frequência.

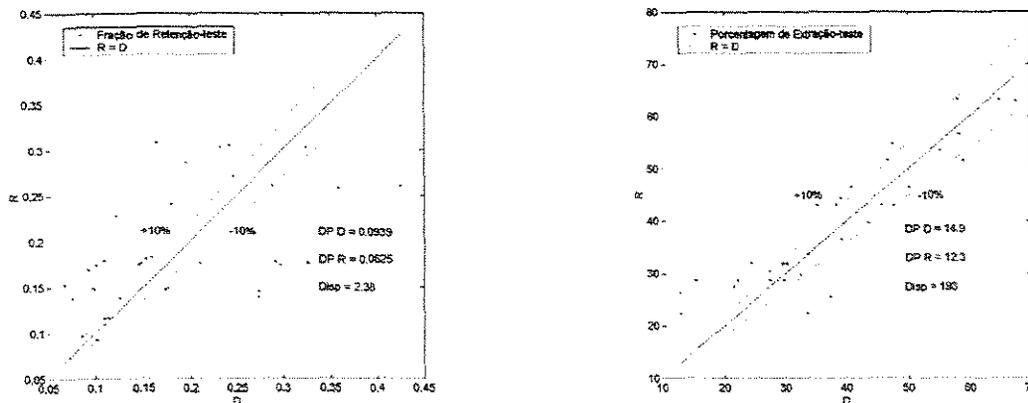


Figura 4.49- Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua com Coeficiente de Partição e Frequência.

Através da análise das figuras 4.48 e 4.49 nota-se que a exclusão da vazão total aumenta a dispersão, sendo que os dados distribuem-se fora da faixa de erro de $\pm 10\%$, concluindo-se que esta variável também é importante para o modelo.

Todas as variáveis inicialmente propostas são importantes para o modelo, concluindo-se que as variáveis de entrada são coeficiente de partição, frequência de pulsação e vazão total das fases.

4.8.2.4 Treinamento do Modelo

Para avaliar-se o desempenho dos algoritmos de treinamento com o conjunto de dados da operação contínua, treinou-se a rede com os mesmos algoritmos anteriores, com dados com variação na entrada e sem variação na entrada, variando-se o número de neurônios da camada intermediária para encontrar o melhor modelo.

4.8.2.4.1 Treinamento com dados com variação na entrada

4.8.2.4.1.1 Algoritmo Levenberg-Marquardt:

Tabela 4.11 - Treinamento com Algoritmo Levenberg-Marquardt.

Levenberg-Marquardt			Conjunto de Treinamento				Conjunto de Teste			
			Fração de Retenção		% de Extração		Fração de Retenção		% de Extração	
Neurônios	Iterações	Des*10 ⁻⁴	Disp	DP R	Disp	DP R	Disp	DP R	Disp	DP R
5	*	12,3	2,78	0,107	673	13,8	0,687	0,102	174	14,3
10	*	5,6	1,56	0,108	430	14,5	0,675	0,0986	123	14,9
15	*	3,5	1,23	0,108	260	14,5	0,652	0,0949	66	17,4
20	*	2,4	0,87	0,108	160	14,5	0,631	0,103	56,8	17,9
22	*	2,2	0,76	0,108	135	14,5	0,681	0,124	73	17,5
24	*	1,4	0,62	0,108	103	14,6	0,78	0,0918	115	17,5
26	*	1,5	0,58	0,108	86	14,6	0,94	0,126	145	17,8
28	*	1,1	0,481	0,108	68	14,6	1,04	0,145	155	22,9
30	*	1,2	0,41	0,108	71	14,6	1,07	0,099	162	17,8
32	606	1,1	0,369	0,108	84,4	14,6	1,13	0,0998	173	17,9

* não alcançou o desempenho desejado até o número máximo de iterações (2000).

Disp – Dispersão dos dados

DP R – Desvio padrão dos dados calculados pela rede

DP R – Desvio padrão dos dados experimentais

Des – Desempenho da rede

Desempenho desejado = 1.10^{-4}

Conjunto de Treinamento

DP D Fração de Retenção = 0,108

DP D % de Extração = 14,6

Conjunto de Teste

DP D Fração de Retenção = 0,0939

DP D % de Extração = 14,9

Através da análise da tabela 4.11 conclui-se que a melhor arquitetura é 3-32-2, devido ser a rede com menor número de neurônios capaz de ajustar os dados alcançando o desempenho desejado (Des).

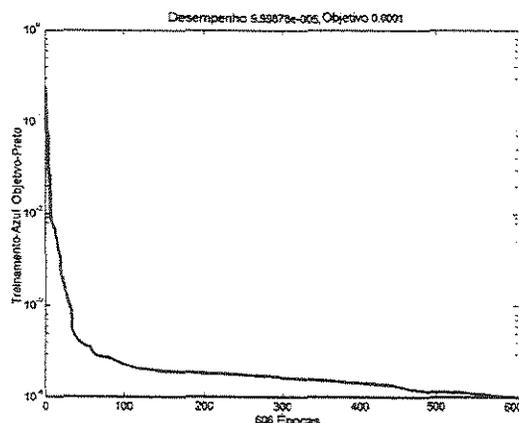


Figura 4.50 - Gráfico do desempenho do Algoritmo Levenberg-Marquardt durante treinamento.

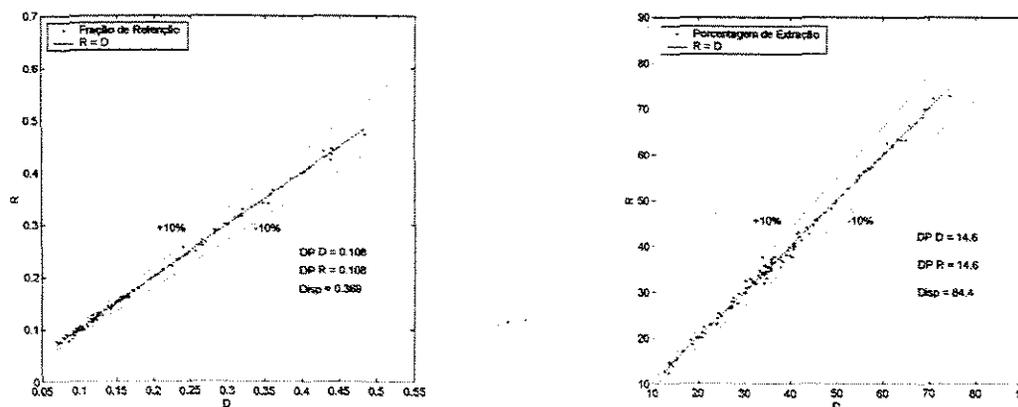


Figura 4.51 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua - com dados com variação na entrada.

Analisando-se os gráficos da figura 4.51 notamos que o conjunto de treinamento obteve uma baixa dispersão e sem dados fora da faixa de erro de $\pm 10\%$, concluindo-se que o treinamento foi satisfatório.

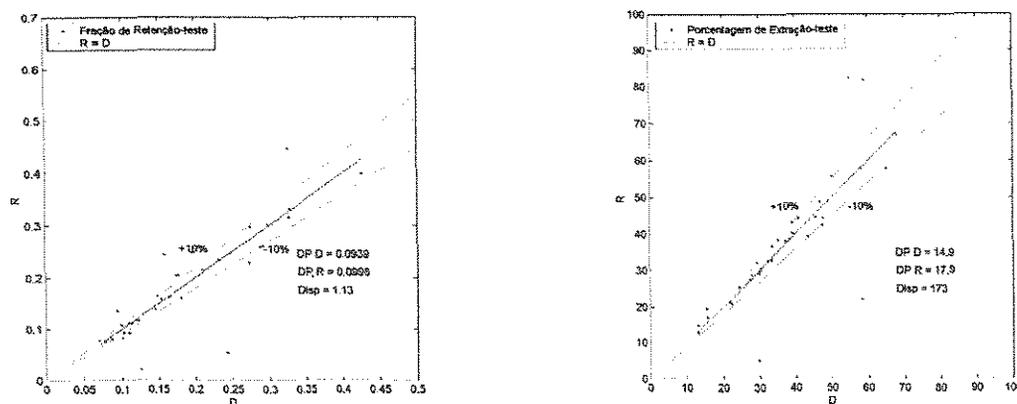


Figura 4.52 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua- com dados com variação na entrada.

Analisando-se a figura 4.52 nota-se que no conjunto de teste a dispersão não é tão baixa e que existem dados fora da faixa de erro de $\pm 10\%$, concluindo-se que houve sobreajuste dos dados. Considerou-se que este modelo não está apto a prever a Fração de Retenção e a Porcentagem de Extração da operação contínua.

4.8.2.4.1.2 Algoritmo Parada Antecipada:

Tabela 4.12 - Treinamento com Algoritmo Parada Antecipada.

Parada Antecipada			Conjunto de Treinamento				Conjunto de Teste			
			Fração de Retenção		% de Extração		Fração de Retenção		% de Extração	
Neurônios	Iterações	Des*10 ⁻⁴	Disp	DP R	Disp	DP R	Disp	DP R	Disp	DP R
2	28	39,5	3,09	0,101	361	13,5	1,11	0,0956	117	14,3
4	239	23,9	1,53	0,104	359	13,8	0,584	0,0956	104	14,8
6	57	9,09	1,42	0,106	179	14,4	0,491	0,096	42,3	14,9
8	48	6,5	0,93	0,106	187	14,4	0,54	0,098	45,1	15,2
10	56	5,7	0,977	0,106	173	14,5	0,544	0,098	42,6	15,3
12	33	5,2	1,08	0,105	150	14,5	0,509	0,092	68	15,8
14	41	5,8	1,23	0,105	182	14,5	0,595	0,092	67,5	16

Disp – Dispersão dos dados

DP R – Desvio padrão dos dados calculados pela rede

DP R – Desvio padrão dos dados experimentais

Des – Desempenho da rede

Desempenho desejado = 1.10^{-4}

Conjunto de Treinamento

DP D Fração de Retenção = 0,106

DP D % de Extração = 14,6

Conjunto de Teste

DP D Fração de Retenção = 0,0939

DP D % de Extração = 14,9

Analisando-se a tabela 4.12 conclui-se que a melhor arquitetura é 3-10-2, devido ser a rede com menor número de neurônios a apresentar baixas dispersões.

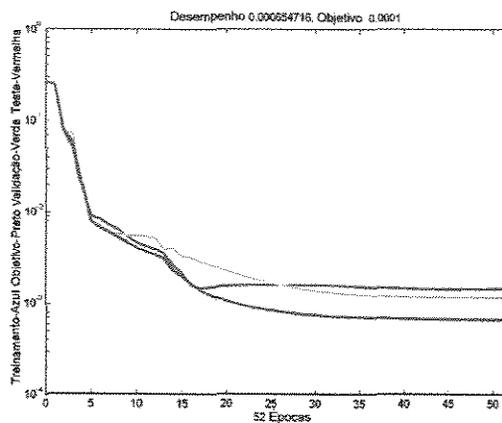


Figura 4.53 - Gráfico do desempenho do Algoritmo Parada Antecipada durante treinamento.

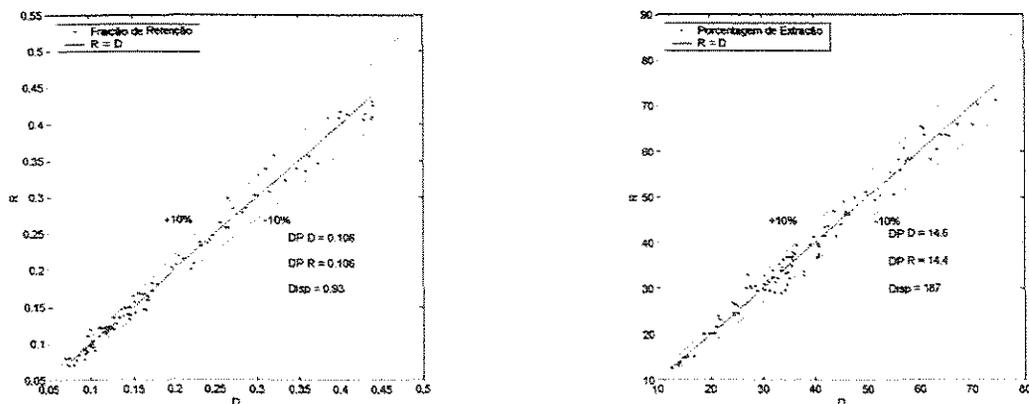


Figura 4.54 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua - com dados com variação na entrada.

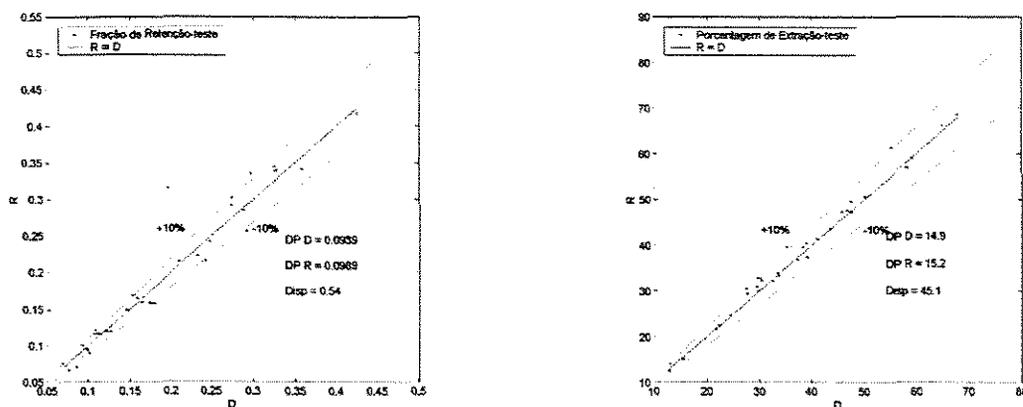


Figura 4.55 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua - com dados com variação na entrada.

Verificando-se as figuras 4.54 e 4.55 nota-se que tanto o conjunto de treinamento quanto o de teste apresentam baixas dispersões, com a maioria dos dados dentro da faixa de erro de $\pm 10\%$, indicando que não houve sobreajuste, obtendo uma melhora significativa em relação ao algoritmo Levenberg-Marquardt o qual apresentou sobreajuste. Este algoritmo ajusta bem mesmo dados com variação na entrada, devido a avaliar o erro do conjunto de validação e parar, caso o erro aumentar, evitando o sobreajuste.

4.8.2.4.1.3 Algoritmo Regularização Bayesiana:

Tabela 4.13 - Treinamento com Algoritmo Regularização Bayesiana.

Regularização Bayesiana					Conjunto de Treinamento				Conjunto de Teste			
					Fração de Retenção		% de Extração		Fração de Retenção		% de Extração	
Neur	Iter.	SQE	SQP	NEP	Disp	DP R	Disp	DP R	Disp	DP R	Disp	DP R
4	120	1,25	3623	23,4	3,6	0,101	5,37	13,5	0,994	0,086	128	13,9
6	189	0,38	13212	33,7	1,87	0,107	350	14,1	0,568	0,098	81,2	14,4
8	239	0,21	7158	45,3	1,5	0,107	247	14,5	0,535	0,099	50,2	15,4
10	274	0,15	5735	55,6	1,2	0,108	198	14,5	0,52	0,1	34,4	15,1
12	412	0,14	5021	66,5	1,15	0,108	185	14,5	0,5	0,098	39,8	15,2
14	385	0,12	5536	74,5	1,14	0,108	165	14,5	0,57	0,099	45,9	15,3
16	829	0,11	5431	69,5	1,01	0,108	161	14,5	0,498	0,099	44,9	15,2
18	673	0,09	5755	94,8	0,828	0,108	162	14,5	0,42	0,098	40,7	15,2
20	1327	0,09	5895	94,1	0,83	0,108	160	14,5	0,49	0,099	50,8	15,7

Disp – Dispersão dos dados

DP R – Desvio padrão dos dados calculados pela rede

DP R – Desvio padrão dos dados experimentais

SQE – Soma quadrática do erro

SQP – Soma quadrática dos pesos

NEP – Número efetivo de parâmetros

Conjunto de Treinamento

DP D Fração de Retenção = 0,108

DP D % de Extração = 14,6

Conjunto de Teste

DP D Fração de Retenção = 0,0939

DP D % de Extração = 14,9

Analisando-se a tabela 4.13 conclui-se que a melhor arquitetura é 3-18-2, devido os parâmetros da rede (SQE, SQP, NEP) ficarem constantes.

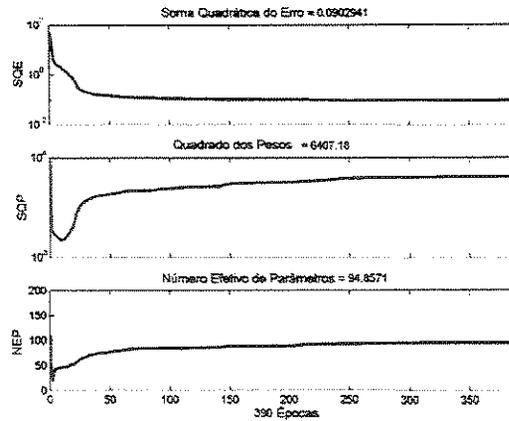


Figura 4.56 - Gráfico do desempenho do Algoritmo Regularização Bayesiana durante treinamento.

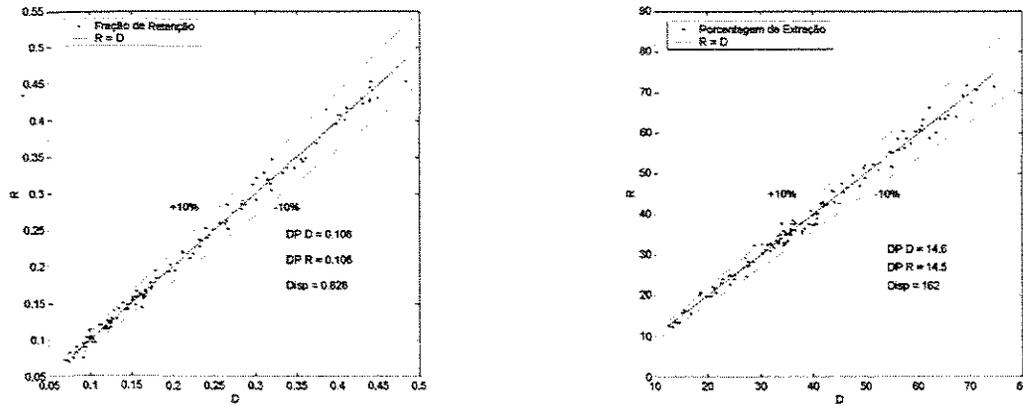


Figura 4.57 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua- com dados com variação na entrada.

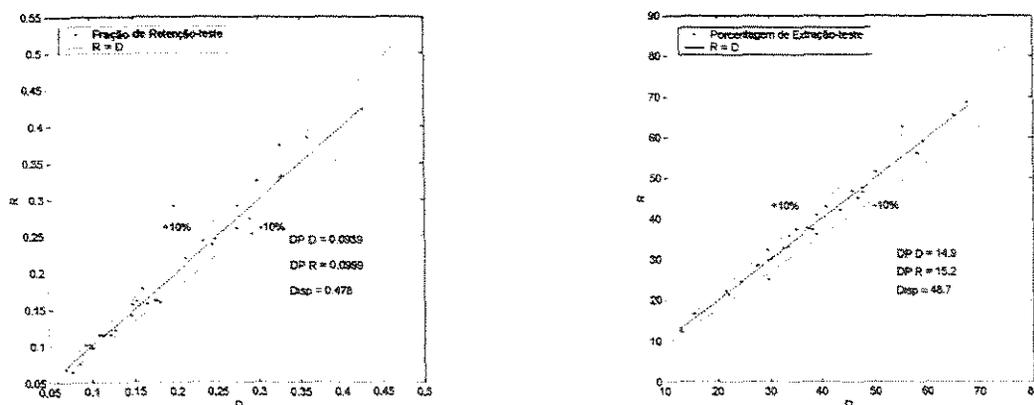


Figura 4.58 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua- com dados com variação na entrada.

Analisando-se as figuras 4.57 e 4.58 notamos que a dispersão do conjunto de treinamento e de teste é baixa e menor que o algoritmo Parada Antecipada, concluindo-se que não houve sobreajuste com um ajuste melhor do conjunto de teste, sendo que isto é devido ao algoritmo de Regularização Bayesiana não separar os conjuntos de treinamento e validação. Então, conclui-se que este modelo está apto a prever a Fração de Retenção e a Porcentagem de Extração para a rede com dados com variação na entrada.

4.8.2.4.2 Treinamento com dados sem variação na entrada

4.8.2.4.2.1 Algoritmo Levenberg-Marquardt:

Tabela 4.14 - Treinamento com Algoritmo Levenberg-Marquardt.

Levenberg-Marquardt			Conjunto de Treinamento				Conjunto de Teste			
			Fração de Retenção		% de Extração		Fração de Retenção		% de Extração	
Neurônios	Iterações	Des*10 ⁻⁴	Disp	DP R	Disp	DP R	Disp	DP R	Disp	DP R
5	*	3,1	2,91	0,105	407	14	1,07	0,0981	12,5	14,58
10	*	3,9	0,779	0,109	166	14,4	0,192	0,0927	54,4	15
15	máx u 266	3,8	0,76	0,109	163	14,4	0,197	0,0922	54	15,1
20	máx u 145	3,8	0,76	0,109	163	14,4	0,197	0,0922	54	15,1
22	máx u 127	3,8	0,76	0,109	163	14,4	0,197	0,0922	54	15,1
24	máx u 261	3,8	0,76	0,109	163	14,4	0,197	0,0922	54	15,1
26	máx u 153	3,8	0,76	0,109	163	14,4	0,197	0,0922	54	15,1
28	máx u 90	3,8	0,76	0,109	163	14,4	0,197	0,0922	54	15,1
30	máx u 119	3,8	0,76	0,109	163	14,4	0,197	0,0922	54	15,1

* não alcançou o desempenho desejado até o número máximo de iterações (2000).

Disp – Dispersão dos dados

DP R – Desvio padrão dos dados calculados pela rede

DP R – Desvio padrão dos dados experimentais

Des – Desempenho da rede

Desempenho desejado = 1.10^{-4}

Conjunto de Treinamento

DP D Fração de Retenção = 0,109

DP D % de Extração = 14,5

Conjunto de Teste

DP D Fração de Retenção = 0,092

DP D % de Extração = 15

Analisando-se a tabela 4.14 observa-se que a rede não chega ao objetivo desejado, devido chegar antes ao mínimo, optou-se então pela menor rede capaz de chegar mais próximo ao desempenho desejado, com isso a melhor arquitetura é 3-15-2.

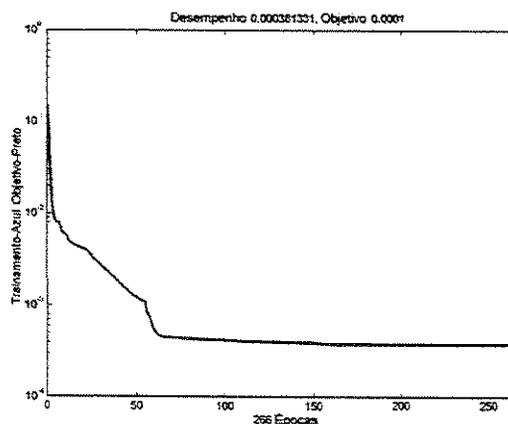


Figura 4.59 - Gráfico do desempenho do Algoritmo Levenberg-Marquardt durante treinamento.

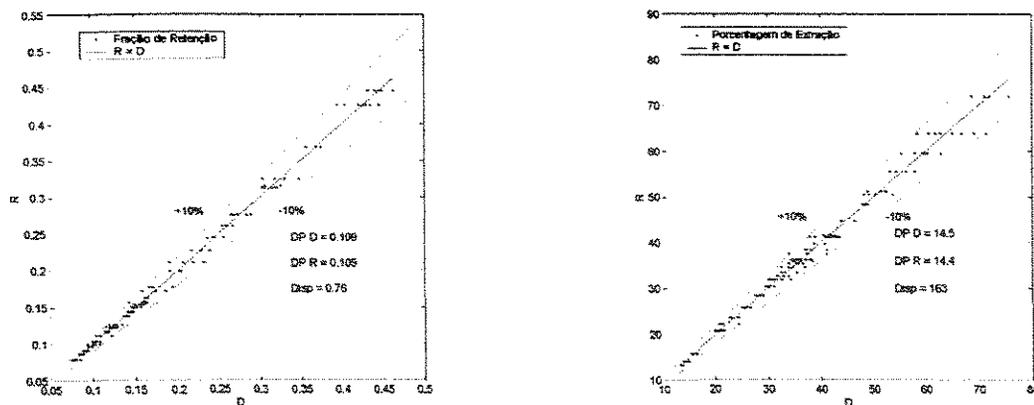


Figura 4.60 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua - com dados sem variação na entrada.

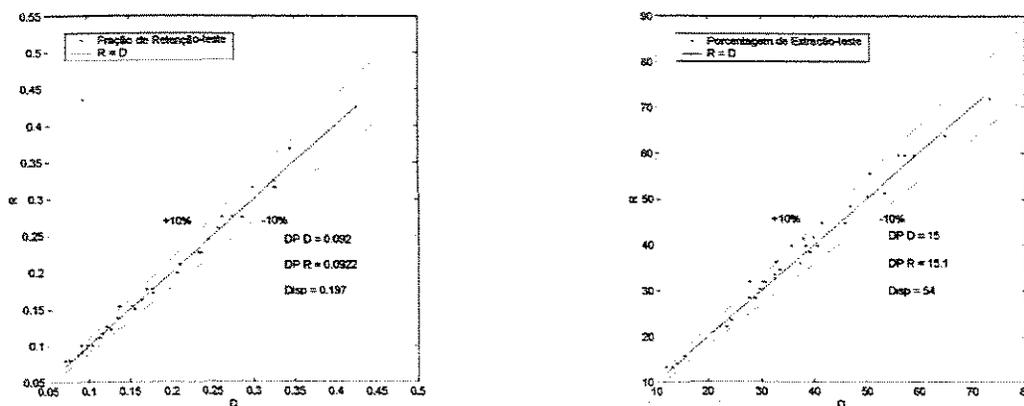


Figura 4.61 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua- com dados sem variação na entrada.

Analisando-se as figuras 4.60 e 4.61 nota-se que a dispersão do conjunto de treinamento é baixa mesmo sem a rede chegar ao objetivo desejado, e com todos os dados dentro da faixa de erro de $\pm 10\%$, o conjunto de teste prova que não houve sobreajuste devido a baixa dispersão. Conclui-se que com os dados de entrada ortogonalizados e com menor dimensão a rede converge para o mínimo, sem sobreajustar.

4.8.2.4.2.2 Algoritmo Parada Antecipada:

Tabela 4.15 - Treinamento com Algoritmo Parada Antecipada.

Parada Antecipada			Conjunto de Treinamento				Conjunto de Teste			
			Fração de Retenção		% de Extração		Fração de Retenção		% de Extração	
Neurônios	Iterações	Des*10 ⁻⁴	Disp	DP R	Disp	DP R	Disp	DP R	Disp	DP R
4	19	25,9	1,89	0,106	369	13,7	0,712	0,092	135	14,2
6	150	11,9	1,51	0,108	177	14,2	0,628	0,092	74,1	14,8
8	82	4,8	0,804	0,109	144	14,3	0,304	0,093	52,8	15,2
10	109	23	1,67	0,11	341	13,6	0,788	0,0917	129	13,9
12	132	9,5	1,49	0,107	200	14,1	0,632	0,095	88,5	14,5
14	82	4,2	0,656	0,109	150	14,3	0,276	0,0937	62,1	15,3

Disp – Dispersão dos dados

DP R – Desvio padrão dos dados calculados pela rede

DP R – Desvio padrão dos dados experimentais

Des – Desempenho da rede

Desempenho desejado = 1.10^{-4}

Conjunto de Treinamento

DP D Fração de Retenção = 0,109

DP D % de Extração = 14,4

Conjunto de Teste

DP D Fração de Retenção = 0,092

DP D % de Extração = 15

Analisando-se a tabela 4.15 a melhor arquitetura é 3-8-2, devido ser a rede que apresenta o desempenho próximo ao desejado, e onde as dispersões são mais baixas no conjunto de treinamento e teste.

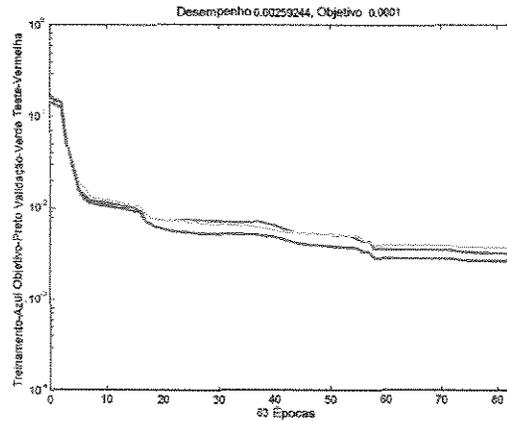


Figura 4.62 - Gráfico do desempenho do Algoritmo Parada Antecipada durante treinamento.

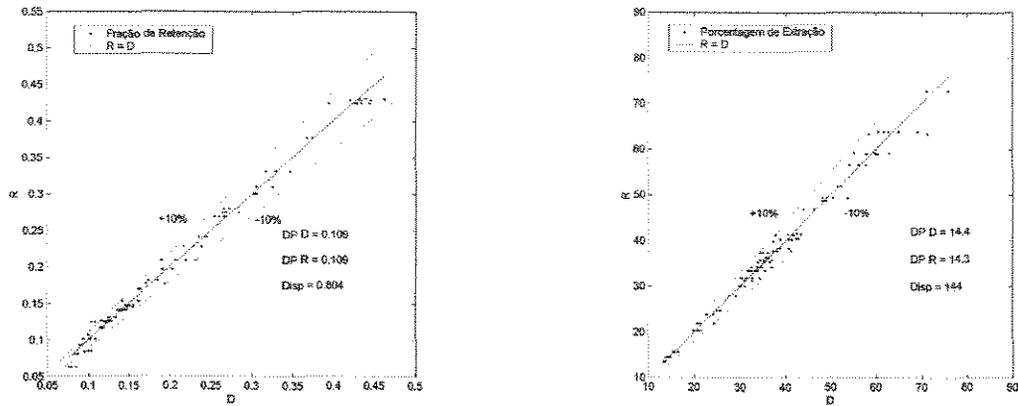


Figura 4.63 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua- com dados sem variação na entrada.

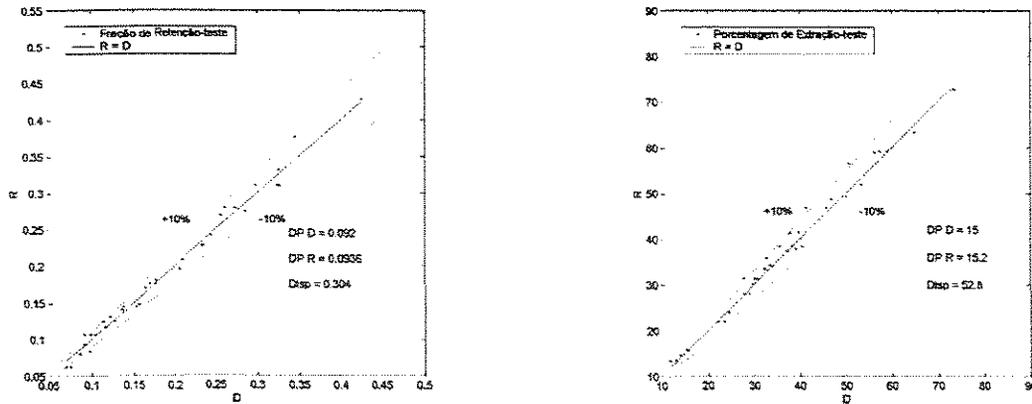


Figura 4.64 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua - com dados sem variação na entrada.

Analisando-se a figura 4.63 nota-se que o ajuste do conjunto de treinamento tem uma dispersão maior que o algoritmo de Levenberg-Marquardt, mas a maioria dos dados encontra-se na faixa de erro de $\pm 10\%$.

Observando-se a figura 4.64 nota-se que mesmo que o conjunto de treinamento tenha apresentado uma maior dispersão, o conjunto de teste apresentou uma dispersão menor, indicando que não houve sobreajuste, sendo que este algoritmo é mais apropriado para o treinamento da rede que o algoritmo anterior.

4.8.2.4.2.3 Algoritmo Regularização Bayesiana:

Tabela 4.16 - Treinamento com Algoritmo Regularização Bayesiana.

Regularização Bayesiana					Conjunto de Treinamento				Conjunto de Teste			
					Fração de Retenção		% de Extração		Fração de Retenção		% de Extração	
Neur.	Iter.	SQE	SQP	NEP	Disp	DP R	Disp	DP R	Disp	DP R	Disp	DP R
4	98	1,63	5691	21,5	3,86	103	708	12,9	1,2	0,093	213	13,1
6	121	0,25	10066	33,1	1,65	0,108	248	14,2	0,472	0,095	81	14,6
8	260	0,14	5654	43,9	0,956	0,109	206	14,3	0,289	0,094	64,4	14,8
10	253	0,12	4753	54	0,917	0,109	176	14,4	0,256	0,094	50,7	15,1
12	279	0,1	4315	60,6	0,811	0,109	163	14,4	0,226	0,092	52,6	15,1
14	949	0,105	3986	61,6	0,792	0,109	164	14,4	0,212	0,092	52,8	15,1
16	741	0,108	3989	62,1	0,816	0,109	167	14,4	0,21	0,093	58,9	15

Disp – Dispersão dos dados

DP R – Desvio padrão dos dados calculados pela rede

DP R – Desvio padrão dos dados experimentais

SQE – Soma quadrática do erro

SQP – Soma quadrática dos pesos

NEP – Número efetivo de parâmetros

Conjunto de Treinamento

DP D Fração de Retenção = 0,109

DP D % de Extração = 14,5

Conjunto de Teste

DP D Fração de Retenção = 0,092

DP D % de Extração = 15

Analisando-se a tabela 4.16 conclui-se que a melhor arquitetura é 3-12-2, devido os parâmetros da rede ficarem constantes (SQE, SQP, NEP), concluindo-se que este é o número de neurônios suficientes.

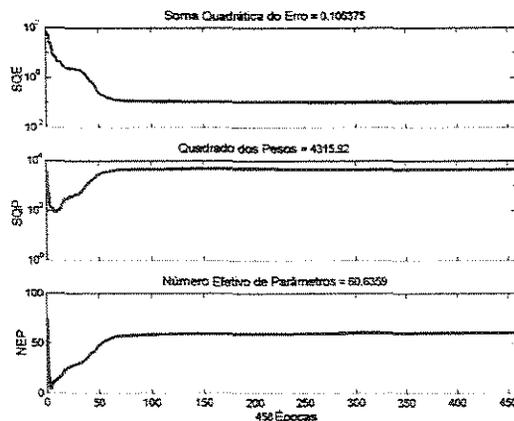


Figura 4.65 - Gráfico do desempenho do Algoritmo Parada Antecipada durante treinamento.

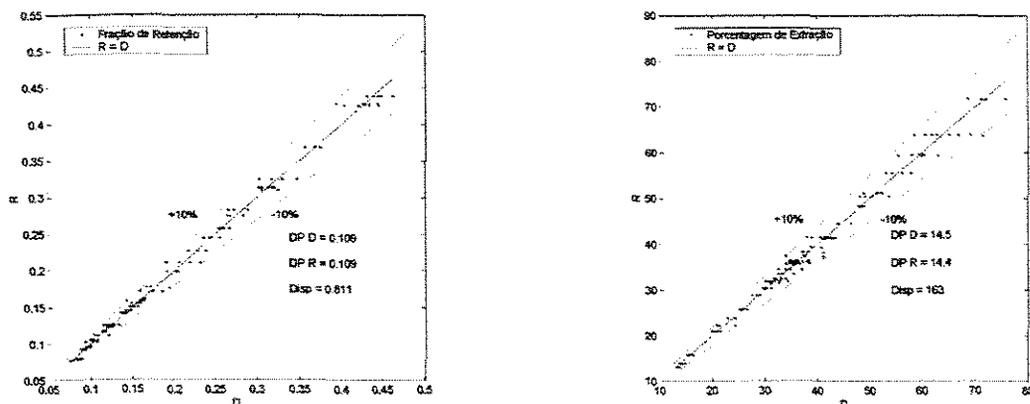


Figura 4.66 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para treinamento da Operação Contínua- com dados sem variação na entrada.

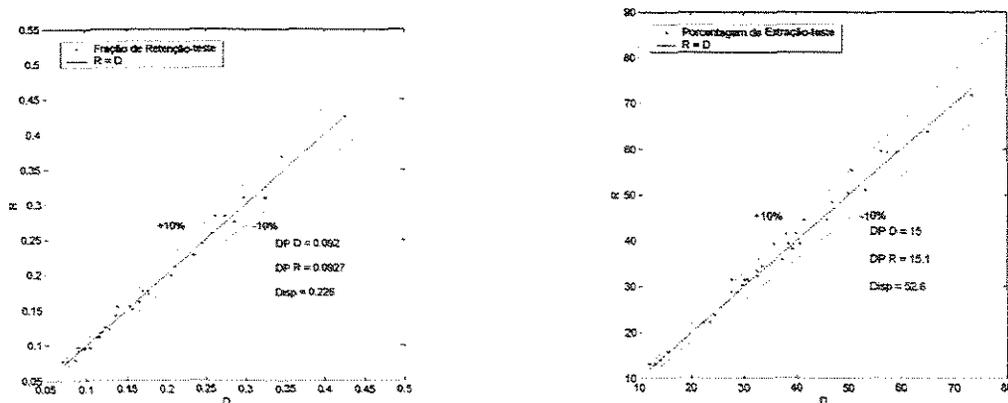


Figura 4.67 - Gráfico dos dados experimentais (D) x dados calculados pela rede (R) para teste da Operação Contínua - com dados sem variação na entrada.

Analisando-se as figuras 4.66 e 4.67 nota-se que ambos conjuntos de teste e treinamento apresentam baixas dispersões, com a maioria dos dados dentro da faixa de erro de $\pm 10\%$, mostrando que a rede chegou ao mínimo da função objetivo e não causou sobreajuste. Com os valores apresentados para a dispersão conclui-se, que este é o melhor algoritmo para a predição da Fração de Retenção e da Porcentagem de Extração com dados sem variação na entrada.

5.0 Conclusões e Sugestões

4.9 Conclusões

Este trabalho teve como objetivo a modelagem via rede neurais da extração do citocromo b5 em operação descontínua e contínua, a fim de se obter um modelo capaz de representar o sistema, dentro de uma faixa de erro da mesma ordem de grandeza dos experimentos.

O modelo que melhor representou a operação descontínua da extração líquido-líquido é o que contém as variáveis de entrada: massa molar do PEG, viscosidade da fase PEG, e viscosidade da fase sal e como saída o coeficiente de partição do citocromo b5. E o modelo que representou melhor a operação contínua é o que contém as variáveis de entrada: coeficiente de partição, vazão total das fases e frequência de pulsação e como saída: porcentagem de extração e fração de retenção da fase dispersa.

Após realizado o trabalho, pode-se concluir que a modelagem via Redes neurais é muito eficiente e pode ser aplicada a outros sistemas.

Algumas conclusões são:

- * A rede neural que apresentou melhor desempenho foi a rede treinada com o algoritmo Regularização Bayesiana.
- * O pré tratamento dos dados de entrada ajuda no treinamento por diminuir a dimensão do vetor de entrada e eliminar dados redundantes.
- * Os melhores algoritmos por ordem são: Levenberg-Marquardt, Parada Antecipada e Regularização Bayesiana.
- * O trabalho com dados experimentais impõe um maior cuidado com a escolha das variáveis e treinamento das redes por conter erros.
- * O algoritmo Regularização Bayesiana tem as melhores respostas mesmo com dados sem tratamento inicial, sendo um bom algoritmo para treinamento de dados experimentais.

As redes neurais demonstraram ser uma excelente ferramenta para a modelagem de sistemas bifásicos com dados experimentais.

4.10 Sugestões

Algumas sugestões para trabalhos futuros são:

- * Uso de outros parâmetros para análise do desempenho das redes, como a distribuição dos dados calculados.
- * Modelagem da extração em sistemas dinâmicos.
- * Execução dos experimentos juntamente com modelagem via redes neurais.
- * Aplicação da modelagem a outros sistemas bifásicos.
- * Modelagem incluindo um maior número de parâmetros (por Ex: pH, ponto isoelétrico, polaridade das moléculas, razão entre as vazões de fases) e aplicável a outros sistemas.

6.0 Referências Bibliográficas

Albertsson, P.A., Partition of Cells Particles and Macromolecules, Wiley- Interscience, New York, 2^a ed 1971.

Albertsson, P.A., Partition of Cells Particles and Macromolecules, Wiley- Interscience, New York, 1986.

Albertsson, P.A.; Carjaville, A.; Brooks, D.E.; Tjerneld, F., Partition of proteins in Aqueous Two-Phase Systems and Effect of Molecular Weight of the Polymer, *Biochem. Biophys. Acta*, 926, 87-93, 1987.

Asenjo, J.A.; Franco, T.T.; Andrews, A.T.; Andrews, B.A., In: *Biological For Recombinant Microorganisms and Animal Cells – New Strategies in Production and Recovery*. Eds. M. White, S. Veuveny and Shafferman. VCH Publisher, 1990.

Bamberger, S.; Brooks, D.; Sharp, K.; Van Alstine, J.; Webber, T., Preparation of Phase Systems and Measurement of Their Physicochemical Properties. In: *Partitioning in Aqueous Two-Phase Systems: Theory, Methods, Uses and Applications to Biotechnology*, eds Walter, H., D.E., Fischer, D., Academic Press, NY, 85-130, 1985.

Baskir, J.N.; Hatton, T.A.; Suter, U.W., Protein partitioning in Two-Phase Aqueous Polymer Systems, *Biotech. Bioeng.*, 34, 541-558, 1989.

Bogdan, S., Gosak D.; Vasić-Racki, D., "Mathematical Modeling of Liquid-Liquid Equilibria in Aqueous Polymer Solutions Containing Neutral Proteinase and Oxitetracycline using Artificial Neural Network", *Computers Chem. Eng.* Vol 19, S791-S796, 1995.

Carlson, A., Factors Influencing the Use of Aqueous Two-Phase Partition for Protein Purification, *Sep. Sci. Technol.*, 23 (8 and 9), 785-817, 1998.

Cascone, O.; Andrews, B.A.; Asenjo, J.A., Partitioning and Purification of Thaumatin in Aqueous Two-Phase System, *Enzyme Microb. Technol.*, 13, 629-635, 1991.

Chouai A., Cabassud M., Lê Lann M.V., Gourdon C., Casamatta G., "Use of neural networks for liquid-liquid extraction column modelling: an experimental study", *Chemical Engineering and Processing* 39, 171-180, 2000.

Coimbra, J.S.R., Desempenho de um Extrator Tipo Graesser na Separação de Proteínas do Soro de Queijo Usando Sistemas Aquosos Bifásicos. Campinas, Tese de Doutorado da Faculdade Engenharia de Alimentos, Universidade Estadual de Campinas, 1995.

Cybenko, G., "Approximations by superpositions of a sigmoidal function", *Math. Control Signal Systems*, 2,303 (1989).

Diamond, A.D.; Hsu, J.T., Aqueous Two-Phase Systems for Biomolecule Separation. In: *Advances in Biochemical Engineering Biotechnology*, Maning (ed), Springer-Verlag Berlin Heidelberg, 47, 90-134, 1992.

Eiteman, M., Prediction Partition Coefficients of Multi-Charged Solut in Aqueous Two-Phase Systems, *J. Chromatography*, 668, 21-30, 1994.

Foresee, F. D., and M. T. Hagan, "Gauss-Newton approximation to Bayesian regularization," *Proceedings of the 1997 International Joint Conference on Neural Networks*, pages 1930-1935, 1997.

Franco, T.T., Use of Modified Proteins in Aqueous Two-Phase Systems. Effect of Surface Hydrophobicity and Charge. Tese de Doutorado. Faculty of Agriculture and Food, University of Reading, 1992.

Giles, A.E.; Aldrich C.; Van Deventer, J.S.J., "Modeling of rare earth solvent extraction with artificial neural nets", *Hydrometallurgy* 43, 241-255, 1996.

Grossman, C.; Zhu, J.; Maurer, G., Phase Equilibrium on Aqueous Two-Phase Systems Containing Aminoacids and Peptides, *Fluid Phase Equilibria*, 82, 275-282, 1993.

Hagan, M.T., Menhaj, M.B., "Training Feedforward Networks with the Marquardt Algorithm", *IEEE Transactions on Neural Networks*, 5(6), 989(1994).

Hanson, C., Recent Research in Solvent Extraction, *Chemical Engineering*, September 9, 135-142, 1968 a.

Hanson, C., Solvent Extraction, *Chemical Engineering*, August 26, 77-98, 1968 a.

Haykin, S., "Neural Networks: a comprehensive foundation", 2nd ed., Prentice Hall, Inc., New Jersey, 1998.

Huddleston, J.G.; Ottomar, K.W.; Ngonyani, D.M.; Lyddiat, A., Influence of System and Molecular Parameters upon Fractionation of Intracellular Proteins from *Saccharomyces* by Aqueous Two-Phase Partition, *Enzyme Microb. Technol.*, 13, 24-32, 1991.

Hustedt, K.H.; Kroner, H.H.; Kula, M.R., Application of Phase Partitioning Biotechnology. In: Walter H., Brooks, D.E. e Fisher, D. (eds), *Partition in aqueous two phase systems*, Academic Press, Orlando, 529-589, 1985.

Johansson, G., Determination of Ionic Charge by Liquid-Liquid Partition, *J. Chromatography*, 32, 425-432, 1985.

Kang, C.H.; Sandler, S.I., A Thermodynamic Model for Two-Phase Aqueous Polymer Systems, *Biotechnol. Bioeng.*, 32, 1158-1164, 1988.

Kawase, Y., Dispersed-Phase Holdup and Mass Transfer in Rotating Disc Contactor with Perforated Skirts, *J. Chem. Tech. Biotechnol.*, 48, 247-260, 1990.

Kim, C.Y.; Brewer, J.W.; Brothers, C.E.; Farver, T.F.; Lee, E.K., Recovery of Extracellular Enzyme from Fermentation Broth by Aqueous Phase Systems, 3^o Chem. Congress of North America, 1998.

Kula, M.R., Trends and Futures Prospects of Aqueous Two-Phase Extraction, *Bioseparation*, 1, 181-189, 1990.

Kumar, A.; Hartland, S., A Unified Correlation for Prediction of Dispersed Phase Hold-Up in Liquid-Liquid Extraction Columns, *Ind. Eng. Chem. Res.*, 34, 3925-3940, 1995.

Kumar, A.; Hartland, S., Independent Prediction of Slip Velocity and Dispersed Phase Hold-Up in Liquid-Liquid Extraction Columns, *Can. J. Chem. Eng.*, 67, 17-25, 1989.

Kumar, A.; Hartland, S., Prediction of Dispersed Phase Holdup in Pulsed Perforated Extraction Columns, *Chem. Eng. Proc*, 23, 41-59, 1988.

Lamarca, C.; Lenhoff, A.M.; Dhurjati, P., Partitioning of Host and Recombinant Cells in Aqueous Two-Phase Polymer Systems, *Biothec. Bioeng.*, 36, 484-492, 1990.

Lehninger, A.L., *Princípios de Bioquímica*. Savier ed., 1986.

Lo, T.C.; Baird, M.H.I., In: *Handbook of Solvent Extraction*. John Wiley & Sons, 1982.

MacKay, D.J.C., "Bayesian Interpolation", *Neural Computation*, vol4, pp.415-447, 1992.

Murphree, E.V., Rectifying Column Calculations- with Particular Reference to N Component Mixtures, *Industrial and Engineering Chemistry*, 17(7), 747-750, 1925.

Murray, R.K.; Graner, D.K.; Mayes, P.A.; Rodwell, P.A., Harper: Bioquímica. Atheneu, São Paulo, 1994.

Papamichael, N.; Boerner, B.; Hustedt, H., Continuous Aqueous Phase Extraction of Proteins: Automated Processing and Recycling of Process Chemicals, *J. Chem. Tech. Biotechnol.*, 54, 47-55, 1992.

Porto, A.L.F., Extração Líquido-Líquido de Proteínas Utilizando Sistemas de Duas Fases Aquosas em Coluna de Discos Perfurados Rotativos. Tese de Doutorado da Faculdade Engenharia Química, Universidade Estadual de Campinas, Campinas, 1998.

Quantrille, T.E., Liu, Y.A., "Artificial Intelligence in Chemical Engineering", Academic Press, Inc., 1991.

Rabelo, A.P.B., Estudo da Eficiência de Separação numa Coluna de Extração Líquido-Líquido Mecanicamente Agitada por Palhetas Rotativas. Tese de Mestrado em Engenharia Química, Faculdade de Engenharia Química, Universidade Estadual de Campinas, Campinas - São Paulo, 1995.

Rabelo, A.P.B., Estudo e Desenvolvimento de uma Micro-Coluna de Campânulas Pulsantes para a Purificação de Proteínas., Tese de Doutorado da Faculdade Engenharia Química, Universidade Estadual de Campinas, Campinas, 1999.

Sarmiento, M.J.; Pires, M.J.; Cabral, J.M.S.; Aires-Barros, M.R., Liquid-Liquid Extraction of a Recombinant rotei, Cytochrome b5, with Aqueous Two-Phase Systems of Polyethylene Glycol and Potassium Phosphate Salts, *Journal of Chromatography A*, 668, 117-120, 1994.

Sebastião, M.J.; Cabral, J.M.S.; Aires-Barros, M.R., Partitioning of Recombinant *Fusarium solani pisi* cutinase in Polyethylene Glycol – Aqueous Salt Solution Two-Phase Systems, *Journal of Chromatography A*, 668, 139-144, 1994.

Simpson, P. K., "Artificial Network Systems", Pergamon Press, New York, NY, 1990.

Szlag, D.C.; Giuliano, K.A., A Low Cost Aqueous Two-Phase System for Enzyme, *Biotech. Techniques*, 4, 277-282, 1998.

Von Bodman, S.B.; Schüller, M.A.; Jollie, D.R.; Slingar, S.G., *Proc Natl. Acad. Sci. USA*, 83 9443-9447, 1986.

Zaslavsky, B.Y.; Borvskaia, A.A.; Gualeva, N.D.; Miheeva, L.M., Physico-Chemical Features of Solvent Media in the Phases os Aqueous Polymer Two-Phase Systems, *Biotech, Bioeng.*, 40, 1-7, 1992.

Zurada, Jacek M., "Introdluction to Artificial Neural Systems", West Publishing Company, United States of America, 1992.