

DANIEL PEREZ SANTANA

**APLICAÇÃO DE REDE NEURAL RECORRENTE DIRIGIDA DINAMICAMENTE EM
PREVISÃO DE FORMAÇÃO DE GEADA**

Dissertação apresentada como requisito parcial
à obtenção do grau de Mestre em Ciências,
Curso de Pós-Graduação em Métodos
Numéricos em Engenharia – Programação
Matemática, Setores de Tecnologia e Ciências
Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Jair Mendes Marques
Co-orientadora: Prof^a Dr^a Maria Terezinha
Arns Steiner

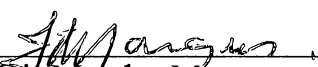
CURITIBA

2003

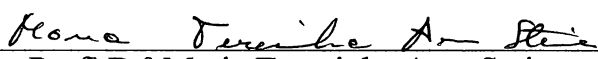
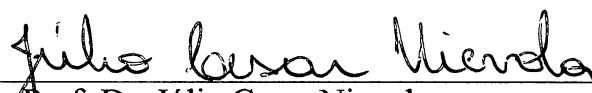
TERMO DE APROVAÇÃO**DANIEL PEREZ SANTANA****APLICAÇÃO DE REDE NEURAL RECORRENTE DIRIGIDA DINAMICAMENTE
EM PREVISÃO DE FORMAÇÃO DE GEADA**

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Ciências no Programa de Pós-Graduação em Métodos Numéricos em Engenharia da Universidade Federal do Paraná, pela comissão formada pelos professores:

Orientador:


Prof. Dr. Jaír Mendes Marques
Departamento de Matemática, UFPR

Co-orientadora:


Profª Drª Maria Terezinha Arns Steiner
Departamento de Matemática, UFPR
Profª Drª L. Valéria R. de Arruda
Programa de Pós-Graduação / CPGEI, CEFET-PR
Prof. Dr. Júlio Cesar Nievola
Coordenador do PPGIA, PUC-PR

Curitiba, 16 de setembro de 2003

DEDICATÓRIA

Dedico em primeiro lugar este trabalho à minha mãe (in memorium) que nunca deixou de dedicar o seu apoio em diversas etapas de minha vida pessoal, profissional e afetiva. Infelizmente, o único momento em que não tive a sua presença a meu lado foi quando ingressei neste mestrado, havia falecido fazia dois meses antes do meu ingresso no curso. Talvez o motivo principal de iniciar o curso foi justamente preencher o meu tempo vago para não lembrar da tristeza desta grande perda.

Dedico também aos meus familiares e amigos que não faltaram com o apoio em todos os momentos difíceis que passei, tanto no âmbito emocional, como no financeiro e no profissional.

AGRADECIMENTOS

Primeiramente a Deus, pela dádiva que me concedeu nas pequeníssimas coisas que surgiram ao longo deste curso. Ingressei neste mestrado a primeira vista para preencher o meu tempo vago para não pensar na perda de minha mãe e, eis que Deus me abençoou com coisas tão preciosas como o apoio especial de quatro pessoas das quais não poderia deixar de mencionar pelos próprios nomes, a saber, as professoras Cleonis, Dayse, Licéia e Luciane; as quais me auxiliaram nos momentos mais difíceis que passei ao longo do curso e das quais tenho uma eterna dívida de gratidão. Outra coisa que parece que também foi providência divina foi constatar que neste curso me deparei justamente com assuntos que tanto me instigavam e interessavam de longa data, tais como as redes neurais.

Em 1998, quando ainda trabalhava em uma indústria petroquímica em Duque de Caxias, R.J., participei na coleta de dados de processo por um período de três meses para uma experiência de otimização de controle de processo que seria executado por um grupo de doutores contratados do M.I.T. que serviria para a criação de uma rede neural que auxiliaria na antecipação de parâmetros ótimos para mudar a produção de uma determinada resina de polipropileno para uma outra sem perdas na qualidade do produto final produzido. Agora, estava vivenciando o conhecimento necessário para compreender o trabalho desenvolvido naquela época.

É digno de mencionar a amizade e auxílio de colega tão abnegado e gentil como o professor Donizetti, sempre prestativo em explicar assuntos tão complexos que nos deparamos durante o Mestrado.

Devo mencionar individualmente a amizade e companheirismo da professora Luciane Brandalise durante o período que tivemos que estudar juntos para a primeira matéria do curso que fizemos após a conclusão das demais matérias, em Curitiba, com o professor doutor Ortega. Agradeço também o apoio de sua irmã, cunhado e irmão caçula que nos cederam hospedagem em seu apartamento no bairro da Água Verde. O apoio do professor doutor Ortega também é digno de nota, sabendo das dificuldades de deslocamento da cidade de Pato Branco para Curitiba, ao sempre atender nossas dúvidas por E-mail e respondê-las prontamente.

Seria injusto de minha parte deixar de mencionar as belas explicações do professor doutor Celso Carnieri e da professora doutora Maria Terezinha Arns Steiner, além da grande dedicação que propiciaram durante os seminários para nos preparar para as nossas defesas da dissertação.

Agradeço ao professor doutor Anselmo que fez avaliar a importância e profundidade da aplicação do estudo estatístico para a fundamentação de dados expressivos e fidedignos para a pesquisa científica.

E, por último, no entanto, tão importante quanto os demais mencionados, o professor doutor Jair Mendes Marques que com sua excepcional serenidade, calma e forma metódica de explicar que

nos fez deveras amar a análise multivariada e avaliar a sua grande significância para este trabalho em questão.

Um agradecimento especial para o Instituto Tecnológico SIMEPAR pelo fornecimento de dados da estação meteorológica de Palmas, P.R., através dos quais se tornou possível a realização deste estudo.

Para não cometer qualquer tipo de injustiça, também presto meus agradecimentos a todos os colegas de curso e pessoas que tornaram possível a realização desta dissertação.

SUMÁRIO

DEDICATÓRIA	iii
AGRADECIMENTOS	iv
SUMÁRIO	vi
LISTA DE FIGURAS	ix
LISTA DE TABELAS	xi
RESUMO	xii
ABSTRACT	xiii
1. INTRODUÇÃO	1
1.1 . PROBLEMA EM FOCO	1
1.2 . OBJETIVO	2
1.3 . ESTRUTURA DA DISSERTAÇÃO	3
2. CONDIÇÕES DE FORMAÇÃO DE GEADA	4
2.1 . MOTIVO PARA O CONGELAMENTO DAS PLANTAS	4
2.1.1 . Nucleação	4
2.2 . CONDIÇÕES CLIMÁTICAS QUE PROVOCAM A GEADA	6
2.2.1 . Comportamento Da Transferência De Calor	6
2.3 . CONDIÇÕES CLIMÁTICAS NO PLANTIO DE MACIEIRAS	8
3. REDES NEURAIS	10
3.1 . INTRODUÇÃO	10
3.2 . A INSPIRAÇÃO BIOLÓGICA	10
3.3 . MODELO ARTIFICIAL BÁSICO	11
3.4 . EMPREGANDO UMA REDE NEURAL	13
Predição em Bolsa de Valores.	13
Controle.	13
3.5 . COLETANDO INFORMAÇÕES PARA A REDE NEURAL	14
3.6 . PRÉ E PÓS PROCESSAMENTO	15
Escalonamento.	16
Valores Nominais	16
3.7 . PERCEPTRONS MULTICAMADA (MLP)	17
3.8 . TREINANDO PERCEPTRONS MULTICAMADA	18
3.9 . O ALGORITMO <i>BACK-PROPAGATION</i>	19
3.10 . SOBRE APRENDIZADO E GENERALIZAÇÃO	20

3.11 . SELEÇÃO DE DADOS	22
O Futuro Não é o Passado.	23
Toda Eventualidade Deve Ser Coberta.	23
A Rede Aprende os Padrões Mais Fáceis.	23
Conjunto de Dados Não Balanceados.	23
3.12 . DETALHES SOBRE TREINAMENTO DE UMA MLP	24
3.13 . OUTROS ALGORITMOS DE TREINAMENTO DE UMA MLP	27
3.14 . CLASSIFICAÇÃO EM REDES NEURAIS	29
Dicotômica.	30
Um de N.	30
3.15 . FOLHA DE DADOS DE CLASSIFICAÇÃO	31
3.16 . PREDIÇÃO EM SÉRIES TEMPORAIS EM REDES NEURAIS	32
3.17 VANTAGENS DE SE EMPREGAR UMA REDE NEURAL RECORRENTE	33
3.18 . ARQUITETURA DE UMA REDE DE ELMAN	34
4 . MATERIAL E MÉTODO	35
4.1 . COLETA DE DADOS	35
4.1.1 . Filtragem Dos Dados	35
4.1.2 . Teste De Normalidade Dos Dados	35
4.1.3 . Normalização Das Variáveis (Z1, Z2,Z3 E Z4)	37
4.1.4 . Verificando A Correlação Entre As Variáveis	38
4.1.4.1 . Correlação entre temperatura mínima e temperatura máxima (x_1 e x_2):	39
4.1.4.2 . Correlação entre temperatura mínima e umidade relativa (x_1 e x_3):	40
4.1.4.3 . Correlação entre temperatura mínima e velocidade do vento (x_1 e x_4):	41
4.1.4.4 . Correlação entre temperatura máxima e umidade relativa (x_2 e x_3):	41
4.1.4.5 . Correlação entre temperatura máxima e velocidade do vento (x_2 e x_4):	42
4.1.4.6 . Correlação entre umidade relativa e velocidade do vento (x_3 e x_4):	42
4.1.4.7 . Conclusões sobre as correlações	43
4.1.5 . Teste de eliminação de variável	44
4.1.6 . Teste de separação e classificação de duas populações	45
4.2 . METODOLOGIA DOS TESTES DE ARQUITETURA DE REDES	47
4.2.1 . Teste das Arquiteturas com 4 Variáveis Não Padronizadas (X_1 , X_2 , X_3 e X_4)	53
4.2.1.1 . Rede de Elman com 4 entradas, 5 neurônios ocultos, função de treinamento “Levenberg-Marquardt” (Elman-X-4-5-LM-nn)	53
4.2.1.2 . Rede de Elman com 4 entradas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptive Learning Rate” (Elman-X-4-5-GDA-0.2-nn)	55

4.2.1.3 . Rede de Elman com 4 entradas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-X-4-5-GDA-2-nn).....	57
4.2.1.4 . Rede de Elman com 4 entradas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-X-4-5-GDA-20-nn).....	58
4.2.1.5 . Rede de Elman com 4 entradas, 10 neurônios ocultos, função de treinamento “Levenberg-Marquardt” (Elman-X-4-10-LM-nn)	60
4.2.1.6 . Rede de Elman com 4 entradas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-X-4-10-GDA-0.2-nn).....	61
4.2.1.7 . Rede de Elman com 4 entradas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-X-4-10-GDA-2-nn).....	63
4.2.1.8 . Rede de Elman com 4 entradas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-X-4-10-GDA-20-nn).....	64
4.2.2 . Teste das Arquiteturas com 4 Variáveis Padronizadas (Z_1 , Z_2 , Z_3 e Z_4).....	65
4.2.2.1 . Rede de Elman com 4 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Levenberg-Marquardt” (Elman-Z-4-5-LM-nn)	65
4.2.2.2 . Rede de Elman com 4 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-4-5-GDA-0.2-nn)	67
4.2.2.3 . Rede de Elman com 4 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-4-5-GDA-2-nn).....	68
4.2.2.4 . Rede de Elman com 4 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-4-5-GDA-20-nn)	69
4.2.2.5 . Rede de Elman com 4 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Levenberg-Marquardt” (Elman-Z-4-10-LM-nn)	71
4.2.2.6 . Rede de Elman com 4 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-4-10-GDA-0.2-nn)	72
4.2.2.7 . Rede de Elman com 4 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-4-10-GDA-2-nn)	73
4.2.2.8 . Rede de Elman com 4 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-4-10-GDA-20-nn)	74
4.2.3 . Teste das Arquiteturas com 3 Variáveis Padronizadas (Z_2 , Z_3 e Z_4).....	75
4.2.3.1 . Rede de Elman com 3 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Levenberg-Marquardt” (Elman-Z-3-5-LM-nn)	75
4.2.3.2 . Rede de Elman com 3 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-3-5-GDA-0.2-nn)	77

4.2.3.3 . Rede de Elman com 3 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-3-5-GDA-2-nn)	78
4.2.3.4 . Rede de Elman com 3 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-3-5-GDA-20-nn)	79
4.2.3.5 . Rede de Elman com 3 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Levenberg-Marquardt” (Elman-Z-3-10-LM-nn)	80
4.2.3.6 . Rede de Elman com 3 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-3-10-GDA-0.2-nn)	81
4.2.3.7 . Rede de Elman com 3 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-3-10-GDA-2-nn)	82
4.2.3.8 . Rede de Elman com 3 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-3-10-GDA-20-nn)	83
4.2.4 . Comparativo dos Melhores Resultados dos Testes	84
4.3 . PÓS-PROCESSAMENTO DA SAÍDA DA REDE.....	85
4.4 . ARQUITETURA FINAL COM OS ESTÁGIOS DE PRÉ E PÓS-PROCESSAMENTO	86
5 . CONCLUSÃO	87
6 . REFERÊNCIA BIBLIOGRÁFICA:	89
APÊNDICE I:	91

LISTA DE FIGURAS

Figura 2.1 Geada em ação no campo	4
Figura 2.2 Nucleação em ação em uma câmara fria	6
Figura 2.3 – gráfico diário da temperatura do ar em diversas alturas acima do solo.....	7
Figura 2.4 – gráfico diário da temperatura do solo em determinadas profundidades.....	7
Figura 2.5 – Fontes da INPE e CPTEC.....	9
Figura 3.1 Diagrama mostrando conexões (sinapses) entre os neurônios.....	13
Figura 3.2 Função de ativação logística (sigmoidal)	16
Figura 3.3 Erro de generalização	21
Figura 3.4 Superfície de resposta sigmoidal para duas entradas	24
Figura 3.5 Curva de resposta típica para uma rede com apenas uma camada de neurônios escondidos.....	25
Figura 3.6 A resposta criada secciona o espaço ao longo de uma diagonal	26
Figura 3.7 Representação esquemática de uma rede de Elman	34
Figura 4.1 Gráfico de normalidade dos dados.....	36

Figura 4.2	Gráfico de dispersão de x_1 e x_2	40
Figura 4.3	Gráfico de dispersão de x_1 e x_3	40
Figura 4.4	Gráfico de dispersão de x_1 e x_4	41
Figura 4.5	Gráfico de dispersão de x_2 e x_3	42
Figura 4.6	Gráfico de dispersão de x_2 e x_4	42
Figura 4.7	Gráfico de dispersão de x_3 e x_4	43
Figura 4.8	Diagrama da rede de Elman com 4 entradas, n neurônios ocultos e um neurônio de saída	49
Figura 4.9	Diagrama da rede de Elman com 3 entradas, n neurônios ocultos e um neurônio de saída	51
Figura 4.10	Desempenho de erro x épocas de treinamento (Elman-X-4-5-LM-03;10)	55
Figura 4.11	Desempenho de erro x épocas de treinamento (Elman-X-4-5-GDA-0.2-04;08)	56
Figura 4.12	Desempenho de erro x épocas de treinamento (Elman-X-4-5-GDA-2-01;03)	58
Figura 4.13	Desempenho de erro x épocas de treinamento (Elman-X-4-5-GDA-20-16;19)	59
Figura 4.14	Desempenho de erro x épocas de treinamento (Elman-X-4-10-LM-11;12)	61
Figura 4.15	Desempenho de erro x épocas de treinamento (Elman-X-4-10-GDA-0.2-03;19)	62
Figura 4.16	Desempenho de erro x épocas de treinamento (Elman-X-4-10-GDA-2-01;02)	64
Figura 4.17	Desempenho de erro x épocas de treinamento (Elman-X-4-10-GDA-20-06;19)	65
Figura 4.18	Desempenho de erro x épocas de treinamento (Elman-Z-4-5-LM-01;12)	66
Figura 4.19	Desempenho de erro x épocas de treinamento (Elman-Z-4-5-GDA-0,2-01;19)	67
Figura 4.20	Desempenho de erro x épocas de treinamento (Elman-Z-4-5-GDA-2-01;05)	69
Figura 4.21	Desempenho de erro x épocas de treinamento (Elman-Z-4-5-GDA-20-07;10)	70
Figura 4.22	Desempenho de erro x épocas de treinamento (Elman-Z-4-10-LM-01;05)	72
Figura 4.23	Desempenho de erro x épocas de treinamento (Elman-Z-4-10-GDA-0.2-07;08)	73
Figura 4.24	Desempenho de erro x épocas de treinamento (Elman-Z-4-10-GDA-2-10;11)	74
Figura 4.25	Desempenho de erro x épocas de treinamento (Elman-Z-4-10-GDA-20-12;19)	75
Figura 4.26	Desempenho de erro x épocas de treinamento (Elman-Z-3-5-LM-12;20)	76
Figura 4.27	Desempenho de erro x épocas de treinamento (Elman-Z-3-5-GDA-0.2-05;18)	77
Figura 4.28	Desempenho de erro x épocas de treinamento (Elman-Z-3-5-GDA-2-09;20)	78
Figura 4.29	Desempenho de erro x épocas de treinamento (Elman-Z-3-5-GDA-20-06;10)	79
Figura 4.30	Desempenho de erro x épocas de treinamento (Elman-Z-3-10-LM-10;17)	80
Figura 4.31	Desempenho de erro x épocas de treinamento (Elman-Z-3-10-GDA-0.2-01;03)	81
Figura 4.32	Desempenho de erro x épocas de treinamento (Elman-Z-3-10-GDA-2-06;20)	82
Figura 4.33	Desempenho de erro x épocas de treinamento (Elman-Z-3-10-GDA-20-03;04)	84

Figura 4.34 Módulos que compõe o autômato de previsão	86
---	----

LISTA DE TABELAS

Tabela 4.1 – Valores de desempenho dos testes (Elman-X-4-5-LM-01 a 20)	54
Tabela 4.2 – Valores de desempenho dos testes (Elman-X-4-5-GDA-0.2-01 a 20)	55
Tabela 4.3 – Valores de desempenho dos testes (Elman-X-4-5-GDA-2-01 a 20)	57
Tabela 4.4 – Valores de desempenho dos testes (Elman-X-4-5-GDA-20-01 a 20)	58
Tabela 4.5 – Valores de desempenho dos testes (Elman-X-4-10-LM-01 a 20)	60
Tabela 4.6 – Valores de desempenho dos testes (Elman-X-4-10-GDA-0.2-01 a 20)	61
Tabela 4.7 – Valores de desempenho dos testes (Elman-X-4-10-GDA-2-01 a 20)	63
Tabela 4.8 – Valores de desempenho dos testes (Elman-X-4-10-GDA-20-01 a 20)	64
Tabela 4.9 – Valores de desempenho dos testes (Elman-Z-4-5-LM-01 a 20)	66
Tabela 4.10 – Valores de desempenho dos testes (Elman-Z-4-5-GDA-0.2-01 a 20)	67
Tabela 4.11 – Valores de desempenho dos testes (Elman-Z-4-5-GDA-2-01 a 20)	68
Tabela 4.12 – Valores de desempenho dos testes (Elman-Z-4-5-GDA-20-01 a 20)	69
Tabela 4.13 – Valores de desempenho dos testes (Elman-Z-4-10-LM-01 a 20)	71
Tabela 4.14 – Valores de desempenho dos testes (Elman-Z-4-10-GDA-0.2-01 a 20)	72
Tabela 4.15 – Valores de desempenho dos testes (Elman-Z-4-10-GDA-2-01 a 20)	73
Tabela 4.16 – Valores de desempenho dos testes (Elman-Z-4-10-GDA-20-01 a 20)	74
Tabela 4.17 – Valores de desempenho dos testes (Elman-Z-3-5-LM-01 a 20)	76
Tabela 4.18 – Valores de desempenho dos testes (Elman-Z-3-5-GDA-0.2-01 a 20)	77
Tabela 4.19 – Valores de desempenho dos testes (Elman-Z-3-5-GDA-2-01 a 20)	78
Tabela 4.20 – Valores de desempenho dos testes (Elman-Z-3-5-GDA-20-01 a 20)	79
Tabela 4.21 – Valores de desempenho dos testes (Elman-Z-3-10-LM-01 a 20)	80
Tabela 4.22 – Valores de desempenho dos testes (Elman-Z-3-10-GDA-0.2-01 a 20)	81
Tabela 4.23 – Valores de desempenho dos testes (Elman-Z-3-10-GDA-2-01 a 20)	82
Tabela 4.24 – Valores de desempenho dos testes (Elman-Z-3-10-GDA-20-01 a 20)	83
Tabela 4.25 – Tabela comparativa dos melhores resultados	84
Tabela 4.26 – Tabela de erros de classificação	85
Tabela I – Valores dos pesos e polarizações finais do teste 10 (Elman-Z-3-5-GDA-20-10)	91
Tabela II – Valores de desempenho do teste 10 (Elman-Z-3-5-GDA-20-10)	91

RESUMO

A geada é um fenômeno baseado no ponto de congelamento do orvalho, o qual pode ocorrer mesmo em temperaturas positivas porque a temperatura do solo pode variar cinco graus Celsius a menos do que a temperatura ambiente local. Algumas regiões sofrem grandes perdas em produção agrícola, principalmente no sul do Brasil e Uruguai, onde a ocorrência da “geada negra” pode atingir as células das plantas e destruir a sua membrana endoplasmática. O objetivo desta dissertação é desenvolver uma rede neural recorrente dirigida dinamicamente para gerar condições de prever a ocorrência deste fenômeno e identificar e classificar os padrões de ocorrência, garantindo condições para uma implementação futura do modelo em um sistema autômato para controlar automaticamente proteções por aspersão de água em macieiras. O trabalho foi todo desenvolvido aplicando-se o pacote de redes neurais do Matlab e depois verificado a sua eficiência.

Palavras-chaves: predição de geada, redes recorrentes dirigidas dinamicamente, rede recorrente simples de Elman.

ABSTRACT

The frost is a phenomenon based in dew freezing which occurs even in positive temperatures because the soil's temperature may acquire a temperature shift over 5 Celsius degrees in comparison with the surroundings. Some regions suffer great damage, mainly the southern part of Brazil and Uruguay, where the "black frost" reaches out the plants' tissues destroying their tissues' membranes. The aim of this work is to develop a recurrent neural network dynamically directed to forecast this phenomenon, identifies and classifies its patterns, guarantying future implementation in an automate system to control automatically protections to spill water onto apple trees to be protected. All the tasks were developed using Matlab's neural network toolbox and afterwards it was verified its certainty.

Keywords: Frost prediction, recurrent networks dynamically directed, Elman's simple recurrent network.

1. INTRODUÇÃO

1.1 . PROBLEMA EM FOCO

A cada ano o superávit da balança comercial brasileira vem provando a importância da agricultura no equilíbrio da balança comercial. Não podemos fugir do fato que apesar do progresso na área industrial, somos um país eminentemente agrícola e temos que explorar estes recursos de maneira inteligente e com baixos custos.

A agroindústria vem provando, em conjunto com pesquisas governamentais, que novos métodos de plantio, tecnologia em equipamentos e melhoria no cruzamento de novos cultivares são necessários para quebras sucessivas de recorde de produção.

No entanto, ainda existem elementos que não se pode controlar, tais como certas condições climáticas adversas que podem provocar grandes perdas de produção.

Um destes fenômenos climáticos é a geada. De acordo com o INPE/CPTEC¹, “geada” é a formação de uma camada de cristais de gelo na superfície ou na folhagem exposta devido à queda de temperatura da superfície abaixo de 0 °C. A principal causa da formação de geada é a advecção de massa de ar polar. Dependendo da intensidade e da extensão da geada, o fenômeno pode causar sérios danos à agricultura, queimando e ressecando a folhagem das plantas, especialmente das hortaliças.

De acordo com a mesma fonte, existem dois tipos de geada:

- Geada “branca” (mais comum) : congela a parte superficial da cultura;
- Geada “negra”: congela a parte interna da cultura;

E as regiões do Brasil mais afetadas pelo fenômeno são:

1. Região Sul
2. Mato Grosso do Sul
3. Oeste e sul do Estado de São Paulo
4. Região da Serra da Mantiqueira (norte de SP e sul de MG)
5. E, ocasionalmente, no sul de Goiás e Triângulo Mineiro

Por outro lado, existem métodos de proteção que podem amenizar as perdas causadas pela geada. De acordo com o CEPAGRI/UNICAMP², o melhor método de proteção direta, caso se concretize a queda de 1 °C/horas é através de irrigação (aspersão, alagamento, regador etc.), com início após a comprovação da queda da temperatura até nível letal. Métodos de proteção práticas tais como cobrir as mudas ou plantas com jornal, sacos de papel ou plásticos também são eficientes.

¹ Informações fornecidas pelo INPE/CPTEC no ano de 1999.

² Informações fornecidas pelo CEPAGRI/UNICAMP no período de Agosto de 1999.

A queima de pneus para fazer fumaça em noite de geada não tem eficiência alguma na proteção das plantas. A fumaça não tem o poder de impedir a perda de calor pelas folhas. A produção de neblina (gotículas de água) é uma forma eficiente de defesa.

Baseado nestas informações, um sistema autômato que monitora as variáveis pertinentes à ocorrência da geada, tais como: temperatura mínima no dia, temperatura máxima no dia, umidade relativa do ar, velocidade do vento, insolação no solo, etc poderia gerar parâmetros de controle que acionariam sprinklers (chuveiros de aspersão) para aspergir uma névoa de água protegendo as plantas antes do fenômeno ocorrer. Contudo, como criar um modelo que empregue tais variáveis e determine o momento exato do acionamento do equipamento de proteção?

1.2 . OBJETIVO

O objetivo deste trabalho é através dos dados fornecidos pelo Instituto Tecnológico SIMEPAR (Estação nº 02651043, Palmas, P.R.) realizar estudos de um modelo para previsão da formação de geada empregando redes neurais.

Através de levantamento de dados realizados nos últimos vinte anos pela estação de Palmas, região sul do Paraná e produtora de maçãs, consistindo de quatro variáveis básicas monitoradas, a saber, temperatura mínima no dia, temperatura máxima no dia, umidade relativa do ar e velocidade do vento, primeiro se fará uma verificação das informações que poderão ser aplicadas para o treinamento e validação do modelo a ser analisado, depois se fará a verificação se a amostra multivariada apresenta um comportamento de distribuição normal.

Depois destas verificações preliminares, um pré-processamento dos dados se faz necessário para que se torne uma amostra padronizada, com média nula e desvio-padrão unitário, em capítulo posterior se demonstrará a justificativa para este tratamento das variáveis de entrada no sistema.

Testes serão executados com o auxílio da “caixa de ferramentas” de redes neurais presente no Matlab 6.0. Com o mesmo pacote será mensurado o tempo de treinamento, a convergência para o erro mínimo definido, a quantidade de épocas de treinamento necessárias para se atingir o alvo.

A partir destes dados, serão feitas comparações estatísticas entre as diversas arquiteturas sugeridas e escolhida a que demonstrou melhor desempenho diante dos parâmetros expostos acima. Far-se-á uma simulação com dados fornecidos pelo SIMEPAR para se verificar a eficiência de resposta da rede.

1.3 . ESTRUTURA DA DISSERTAÇÃO

O capítulo 2 da dissertação apresenta as justificativas científicas para se realizar um trabalho de proteção contra a geada, apresentando primeiro uma teoria básica de botânica e princípios de termodinâmica que explicam o comportamento do fenômeno. Também se apresenta a condição climática para a identificação das variáveis pertinentes a criação do modelo.

O capítulo 3 demonstra um resumo teórico sobre redes neurais, funcionamento básico, tipos de arquiteturas e as aplicações mais utilizadas para cada modelo. Também se apresentam as justificativas para se realizar o pré-processamento dos dados de treinamento e uma explicação sucinta sobre o algoritmo de aprendizagem.

No capítulo 4 apresentam-se as explicações de todos cálculos realizados, tabelas com resumo das comparações entre as diversas arquiteturas empregadas, com auxílio dos anexos que apresentam na íntegra os relatórios gerados pelo pacote do Matlab e de gráficos de desempenho. A conclusão define o modelo escolhido, seu comportamento e eficiência, além de proposta de melhorias sugeridas para a continuação deste trabalho.

No capítulo 5 apresentam-se as conclusões e sugestões para melhorias.

2. CONDIÇÕES DE FORMAÇÃO DE GEADA

A geada causa danos consideráveis às colheitas ao redor do mundo. A causa principal do dano é o resfriamento excessivo da superfície da planta e congelamento subsequente das células das plantas devido ao fluxo de calor por irradiação que sai da planta em direção à atmosfera.

Isto ocorre geralmente durante noites de céu limpo, sem a presença de vento, sem nuvens e com umidade relativa do ar baixa³.

Figura 2.1 Geada em ação no campo



Fonte: INPE/CEPTEC

De maneira a se evitar prejuízos para as plantas devido à geada, diversos métodos de proteção são empregados tais como: cortina de neblina, molhar o solo e as plantas, técnicas de cobertura utilizando plástico, papel ou espuma.

2.1 . MOTIVO PARA O CONGELAMENTO DAS PLANTAS

As plantas congelam quando não podem evitar a nucleação (veja a seguir) e prevenir-se do crescimento de partículas de gelo. No campo, por exemplo, os pessegueiros sofrem nucleação entre temperaturas de -0,6 e -2,6 °C enquanto outras espécies de clima mais temperado sofrem o fenômeno entre -1,2 e -2,1 °C (Ashworth e Davis, 1986) e gramíneas podem congelar entre -1,5 °C e -2,5 °C (Pearce e Fuller, 2001).

2.1.1 . Nucleação

É quando as moléculas de água se agrupam para formar um núcleo de gelo estável, tanto espontaneamente, o que se chama nucleação homogênea, ou quando catalisado por uma outra

³ Informações fornecidas pelo CEPAGRI/UNICAMP no período de Agosto de 1999

substância qualquer, o que caracteriza uma nucleação heterogênea. A nucleação homogênea ocorre somente em temperaturas muito baixas, em contraste, em climas úmidos, a nucleação heterogênea ocorre mais facilmente em temperaturas menos baixas.

A quantidade de água pura necessária para gerar um núcleo estável de gelo através de nucleação homogênea é alta: $2,3 \times 10^{312}$ kg em $-10\text{ }^{\circ}\text{C}$ (Franks, 1985). Esta quantidade decai conforme a temperatura reduz, mas continua alta em $-30\text{ }^{\circ}\text{C}$ ($3,8 \times 10^{11}$ kg). A quantidade vai diminuindo até atingir a temperatura limite de $-38,5\text{ }^{\circ}\text{C}$ onde qualquer volume de água congela (Franks, 1985).

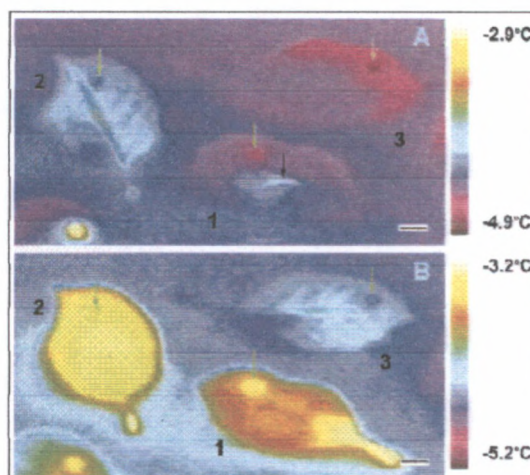
Contudo, existem substâncias na natureza que podem agir como nucleadores heterogêneos tais como:

- a) bactérias de nucleação ativa (INA – ice nucleation-active);
- b) outras moléculas biológicas e estruturas;
- c) resíduos orgânicos e inorgânicos.

Os nucleadores se encontram tanto extrinsecamente na superfície da planta ou, em alguns casos, internamente na planta (intrinsecamente). Para entrar em ação, um nucleador heterogêneo em potencial deve entrar em contato com água. Conseqüentemente, se a superfície da planta estiver seca, os nucleadores estarão inativos. Contudo, durante uma geada em muitos climas, a umidade se condensa na superfície da planta o que efetiva a oportunidade dos nucleadores heterogêneos presentes serem ativados.

Existem plantas que possuem nucleadores que não contém bactérias como no caso da espécie Afro-alpina que se baseia em polissacarídeos que provocam o congelamento acima de $-4\text{ }^{\circ}\text{C}$ (Griffith e Antikien, 1996). Na **figura 2.2** verifica-se através de fotografia por infravermelho durante um congelamento controlado em uma câmera fria (Atherton, Pearce e Fuller, resultados não publicados e métodos descritos por Pearce e Fuller, 2001) a nucleação se espalhando através das folhas já que o congelamento trata-se de um efeito exotérmico e que pode ser detectado por infravermelho.

Figura 2.2 Nucleação em ação em uma câmara fria



Fonte: Department of Biological and Nutritional Sciences, The University of Newcastle, U.K.

2.2 . CONDIÇÕES CLIMÁTICAS QUE PROVOCAM A GEADA

As condições básicas pesquisadas nos últimos anos apontam para os seguintes fatos: noite límpida (estrelada), sem vento, sem nuvens, regiões montanhosas ou regiões relativamente elevadas, depressões (vales) em regiões montanhosas: nestas regiões formam-se "lagos" de ar frio devido ao acúmulo de ar que desce as encostas vizinhas e com umidade do ar baixa.

2.2.1 . Comportamento Da Transferência De Calor

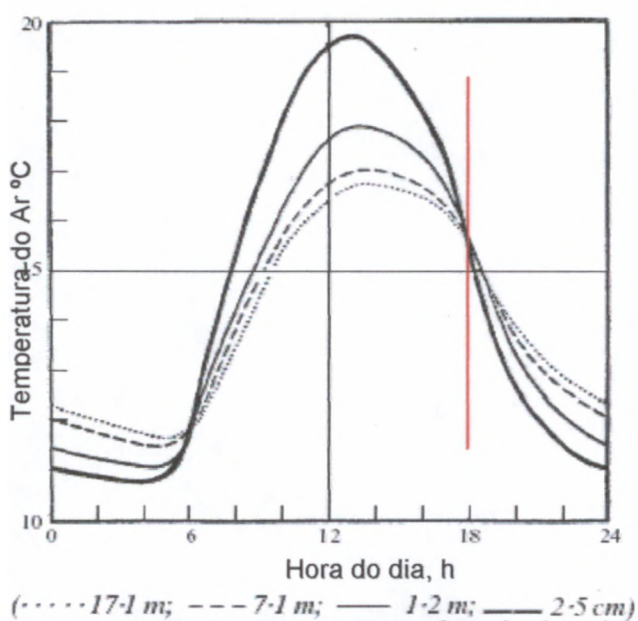
Durante a noite, as camadas superiores do solo se esfriam devido a perdas de calor por irradiação de infravermelho saindo da própria superfície. Esta perda é derivada da diferença entre a irradiação Stefan-Boltzmann vinda da superfície do solo e a contra irradiação vinda da atmosfera, a qual dependem da temperatura do ar, da quantidade de vapor d'água presente, de dióxido de carbono, ozônio e nebulosidade. Já que a proporção de dióxido de carbono e ozônio na atmosfera é considerada constante, a intensidade de perda por irradiação noturna depende essencialmente da quantidade de vapor d'água e temperatura do ar. Os fluxos de calor entre as camadas superiores do solo e do ar adjacente também contribuem para o balanço de calor da superfície do solo.

No início de uma noite de geada, a temperatura do ar e a temperatura vertical do solo próximo à superfície do solo podem ser presumidos como uniforme. Este fenômeno está relacionado à inversão do perfil de temperatura que ocorre um pouco antes ou depois de um pôr-do-sol. Uma confirmação desta avaliação se obtém através de medições das temperaturas do ar e da superfície demonstrando uniformidade na distribuição de temperatura no ar adjacente à superfície do solo e nas camadas superiores do solo (figuras a seguir). Como se destaca nas figuras 3 e 4, a distribuição

de temperatura da camada superior do solo e da camada inferior de ar durante a inversão, que ocorre aproximadamente às 18:00 horas, são uniformes. Deve-se enfatizar que estas medições estão relacionadas a experimentos diferentes e, portanto, os valores das temperaturas de terra e ar no momento da inversão são diferentes.

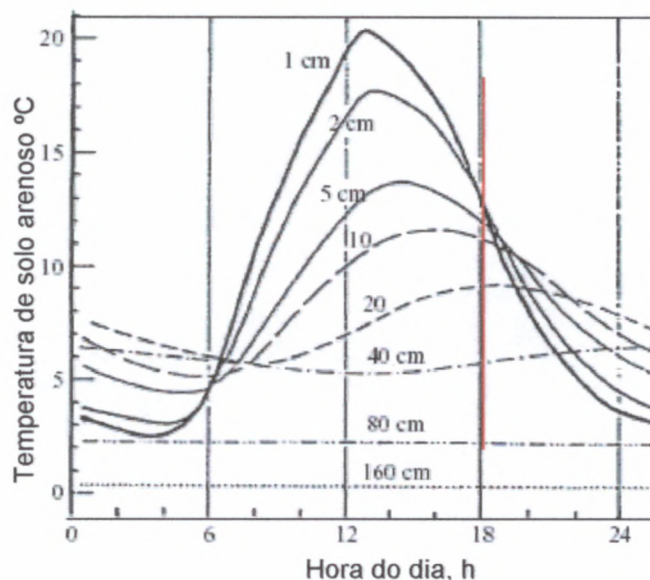
Ao se efetuar a medição da temperatura do ar e do solo no período do pôr-do-sol, é possível se criar um modelo de função de transferência de calor unidirecional baseado na constante de irradiação de Stefan-Boltzmann, levando-se também em consideração parâmetros de condutibilidade térmica do solo e do ar e conseguir prever se a temperatura ao nível do solo chegará ao longo da madrugada a um valor limite aonde a nucleação heterogênea ou geada possa provocar danos às plantas (B. Krasovitski ; E. Kimmel ; I. Amir, 1995).

Figura 2.3 – gráfico diário da temperatura do ar em diversas alturas acima do solo.



Fonte: Department of Agricultural Engineering, Technion, Haifa 32000, Israel

Figura 2.4 – gráfico diário da temperatura do solo em determinadas profundidades.



Este método só pode ser reproduzido em laboratório sob condições controladas e apropriadas, porém, apresenta dificuldades para ser executado em termos práticos gerais.

Para isto o modelo necessita de parâmetros que são difíceis de se determinar por serem valores médios dependentes do tempo e do espaço. Por exemplo, as propriedades de transferência de calor do solo são afetadas pelo tipo do solo e pelo teor de umidade do mesmo que varia substancialmente ao longo das estações do ano (tempo), dependendo das condições climáticas e do local em estudo (espaço).

A princípio, isto não se torna um estorvo se as medições destas características do solo em estudo sejam feitas ao longo do ano, mas existe um outro fator crucial que complica este método, que tipo de sensoriamento se emprega para detectar que a noite está estrelada, já que este fenômeno está diretamente atrelado a esta condição?

A seguir, nesta dissertação, apresentamos uma proposta de nível mais prático para resolvermos a questão de previsão da geada, no momento vamos analisar outros fatores importantes para justificar a aplicação prática na agricultura desta previsão.

2.3 . CONDIÇÕES CLIMÁTICAS NO PLANTIO DE MACIEIRAS

O plantio de macieiras depende de condições climáticas bastante especiais devido às características da planta. As macieiras possuem uma produção maior de frutos e com uma qualidade superior quando se desenvolvem em um clima relativamente frio. Contudo, isto gera um dilema, enquanto o fruto maduro precisa de condições de frio para se desenvolver e produzir uma maior quantidade de frutose, o mesmo frio pode propiciar as condições necessárias e suficientes para produzir a geada que pode queimar as flores e os frutos novos.

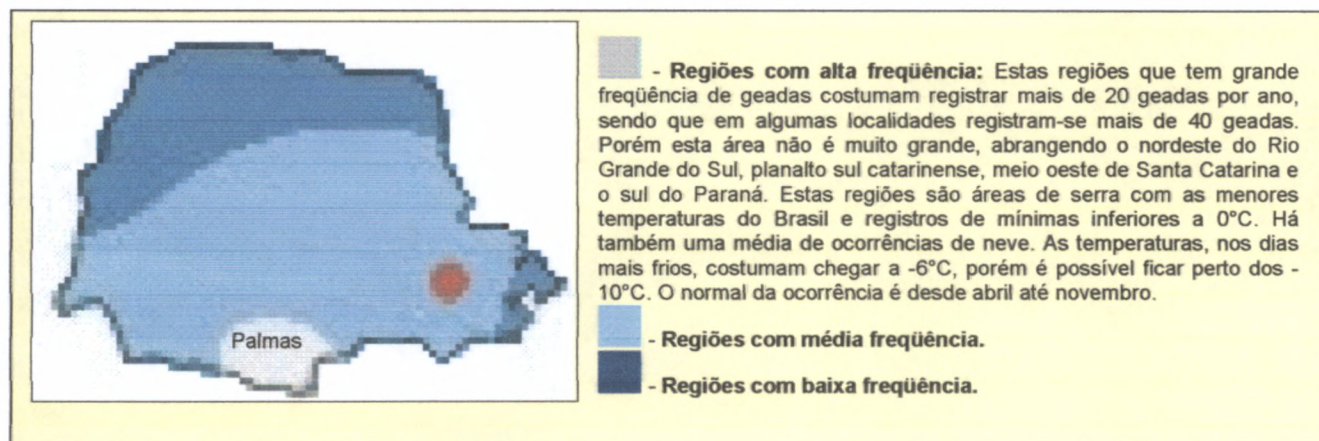
Corroborando a acima exposta, basta verificar no mapa do Paraná que a região mais propícia para a produção de maçãs, a cidade de Palmas, corresponde ao ponto mais frio e à região de maior incidência de geada no estado.

O objetivo desta dissertação é propor uma forma de se criar um sistema de proteção das macieiras automatizado empregando-se microcontroladores e aplicando-se redes neurais para se modelar o algoritmo de controle de sprinklers para aspersão de água sobre os pomares.

O questionamento é: como a aspersão de água não provoca o fenômeno da geada sobre os pomares? A resposta está no equilíbrio térmico obtido, enquanto que com a umidade presente no ar ocorre a nucleação heterogênea porque certos componentes extrínsecos e intrínsecos nas plantas acabam "atraindo" estas moléculas de água, cristalizando o meio aquoso das células e que ao se aquecer para se evaporar, provoca a expansão da célula e o rompimento da membrana celular, é o efeito chamado popularmente de "queima da planta". Já a aspersão de água provoca a saturação e o congelamento da água que ocorre externamente na planta, formando uma camada que se mantém a 0 °C, diferente da nucleação que ocorre em valores negativos, quando ocorre o aquecimento natural com o nascer do sol a temperatura se mantém em 0 °C até se transformar toda a camada de gelo em água e passa a se elevar lentamente até atingir a temperatura ambiente.

Com este método, protege-se a planta da incidência de cristais de gelo formado em temperaturas negativas por se formar um invólucro de temperatura superior e que não afeta as células da planta.

Figura 2.5 – Fontes da INPE e CPTEC



Outro questionamento na aplicação deste método de proteção é o momento exato para se disparar os sprinklers e como automatizá-los. É neste contexto que se concentra o objetivo desta dissertação, a implementação prática de um sistema automático.

A melhor opção passou a ser a utilização de um sistema baseado em rede neural por ter a capacidade de ser empregado na identificação de padrões que apontariam para a iminência de geada. Existem diversos tipos de arranjo de redes neurais e o que se apresenta mais propício para este caso será exposto a seguir.

3. REDES NEURAIIS

3.1 . INTRODUÇÃO

As redes neurais têm experimentado uma explosão de interesses através dos últimos anos, sua aplicação abrange uma grande área de atividades tais como finanças, medicina, engenharia, geologia e física. De fato, em qualquer área que existam problemas de predição, classificação ou controle, se pode empregar redes neurais. Este sucesso é atribuído a diversos fatores chaves:

- **Poder de Resolução.** As redes neurais são técnicas bastante sofisticadas de modelamento capazes de modelar funções extremamente complexas. Em particular, as redes neurais são não-lineares. Por muitos anos o modelamento linear têm contribuído na maioria das técnicas de modelagem, porque tais modelos possuem excelentes estratégias de otimização. Contudo, aonde não for possível a aproximação linear (no qual se enquadra a maioria dos casos na prática) tais modelos não levam a bons resultados. As redes neurais colocam também em prova a *maldição da dimensionalidade* a qual frustra qualquer tentativa de se linearizar qualquer função não-linear com um grande número de variáveis (geralmente se empregando multiplicadores de Lagrange).
- **Fácil Utilização.** As redes neurais aprendem por exemplo. O usuário de redes neurais reúne dados representativos e então realiza *algoritmos de treinamento* para o aprendizado automático da estrutura de dados. Embora o usuário necessite ter algum conhecimento heurístico de como selecionar e preparar os dados, em como selecionar uma rede neural apropriada, e como interpretar os resultados, o nível de conhecimento do usuário necessário para se aplicar com sucesso à rede neural é inferior ao empregado em métodos estatísticos tradicionais.

As redes neurais também são intuitivas por se basearem no modelo básico dos sistemas neurais biológicos.

3.2 . A INSPIRAÇÃO BIOLÓGICA

As redes neurais foram desenvolvidas a partir da pesquisa em Inteligência Artificial que, especificamente, pretende imitar os sistemas neurais biológicos na capacidade de aprendizado e na tolerância às faltas ao se criar um modelo do sistema básico do cérebro (Patterson, 1996). Os principais grupos de pesquisa em Inteligência Artificial nas décadas de 60 a 80 produziram Sistemas Especialistas. Estes sistemas são baseados em modelos de alto nível no processamento do raciocínio (especificamente, o conceito de que o raciocínio é construído a partir da manipulação de

símbolos). Contudo, verificou-se que tais sistemas falham em capturar certos aspectos da inteligência humana. De acordo com algumas referências isto ocorre por causa da falha em se imitar a camada inferior do cérebro já que para se reproduzir a inteligência seria necessário construir sistemas com uma arquitetura similar.

O cérebro é composto principalmente de um grande número de neurônios (aproximadamente 10.000.000.000) que se conectam de maneira massiva (com uma média de centenas de interconexões por neurônio, embora esta cifra possa variar amplamente). Cada neurônio é uma célula especializada que pode propagar sinais eletroquímicos. O neurônio possui uma estrutura de entrada de dados (os dendritos), um corpo celular e uma estrutura ramificada de saída (o axônio). Os axônios de uma célula se conectam aos dendritos de outras células através de uma sinapse. Quando um neurônio é ativado, ele dispara um sinal eletroquímico através do axônio. Este sinal cruza as sinapses para outros neurônios, o qual por sua vez podem disparar seus próprios sinais. Um neurônio só dispara se o sinal total recebido no corpo celular através dos dendritos exceder um certo valor (limite de disparo).

A potência do sinal recebido por um neurônio (e portanto a sua chance de disparar) depende da eficácia das sinapses. Cada sinapse contém uma barreira que permite a química neuro-transmissora transmitir um sinal através desta barreira. Um dos mais importantes pesquisadores em sistemas neurológicos (Donald Hebb) postulou que a aprendizagem consiste principalmente em se alterar a “potência” das conexões sinápticas. Por exemplo, no experimento clássico de condicionamento de Pavlov, aonde o sino é tocado antes da refeição ao se dar a comida para um cão, o cão aprende rapidamente a associar o sino com o ato de comer. As conexões sinápticas entre a parte apropriada do córtex cerebral e as glândulas salivares são fortalecidos, de maneira que o córtex é estimulado pelo som do sino a fazer com que o cão salive.

Desta forma, através de um grande número de unidades de processamento extremamente simples, cada um realizando uma soma ponderada de suas entradas, e então disparando um sinal binário se um valor de entrada total exceder certo valor, o cérebro manipula tarefas extremamente complexas.

3.3 . MODELO ARTIFICIAL BÁSICO

Para se capturar a essência de um sistema neural biológico se define um neurônio artificial conforme a seguir:

- Recebe um número de entradas (tanto de dados originais ou da saída de outros neurônios em uma rede neural). Cada entrada vem através de uma conexão que possui uma potência (ou peso); estes pesos correspondem a eficácia da sinapse em um neurônio biológico. Cada

neurônio também possui um valor limite unitário. A soma ponderada das entradas é formada, e o limite subtraído, para compor a *ativação* do neurônio também conhecido como Potencial Pós-Sináptico ou PSP (*“Post-Synaptic Potential”*) do neurônio.

- O sinal de ativação é passado através de uma *função de ativação* também conhecido como uma função de transferência para produzir a saída do neurônio.

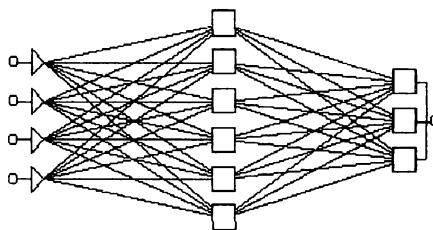
Se for empregado a função de ativação degrau unitário então o neurônio atua da mesma forma que o neurônio biológico descrito anteriormente (subtraindo o limite da soma ponderada e comparando com zero o que é equivalente a se comparar a soma ponderada ao limite). Atualmente, a função degrau unitário é raramente empregada em rede de neurônios artificiais, conforme será discutido em breve. Note que o peso pode ser negativo, no que implica que a sinapse possui um efeito inibitório em vez de um efeito excitatório no neurônio: neurônios inibitórios também são encontrados no cérebro.

Como os neurônios devem ser conectados juntos? Se a rede deve ser útil para qualquer utilização, deve haver entradas que carregam os valores de variáveis de interesse do mundo exterior e saídas que formam predições, ou sinais de controle. As entradas e as saídas correspondem aos nervos sensores e motores tais como aquelas vindas dos olhos e que comandam as mãos. Contudo, pode haver também neurônios escondidos que exercem tarefas internas na rede.

A solução chave é realimentação ou *feedback* (Haykin, 2001). Uma rede simples possui uma estrutura de retroalimentação ou *feedback*: os sinais fluem através das entradas, se encaminha pelas unidades escondidas e alcançam as unidades de saída. Tal estrutura possui comportamento estável. Contudo, se a rede for *recorrente* (contém conexões entre as saídas e os neurônios anteriores ou seja são realimentados) pode se tornar instável e possuir uma dinâmica muito complexa. Redes recorrentes são de grande interesse para pesquisadores em redes neurais, no entanto têm sido as estruturas do tipo *feedback* que se têm provado útil em solucionar certos problemas reais.

Mostramos a seguir uma rede *feedforward* típica. Os neurônios são dispostos em uma topologia em camadas distinta. A camada de entrada não é composta de neurônios: estas unidades simplesmente servem para introduzir os valores das variáveis de entrada. Os neurônios das camadas escondidas e de saída são todos conectados com todas as unidades da camada precedente. Novamente, é possível se definir redes que são conectadas parcialmente com algumas unidades da camada precedente; mas para muitas aplicações é melhor se empregar redes com conexões completas.

Figura 3.1 Diagrama mostrando conexões (sinapses) entre os neurônios



Fonte: StatSoft's Electronic Book Homepage

Quando a rede for executada, os valores das variáveis de entrada são aplicadas nas unidades de entrada, e então as unidades das camadas escondidas e de saída são progressivamente executadas. Cada uma dessas unidades calcula o seu valor de ativação por pegar a soma ponderada das saídas das unidades da camada precedente e subtrair o limite. O valor da ativação é passado através da função de ativação para produzir a saída do neurônio.

Quando a rede inteira for executada, as saídas da camada de saída atuam como a saída da rede inteira.

3.4 . EMPREGANDO UMA REDE NEURAL

O tipo de problema possível de ser solucionado por uma rede neural se define pela maneira de *trabalho* e pela maneira de *treinamento*. As redes neurais trabalham a partir de variáveis de entrada fornecidas como alimentação de dados e produzindo algumas variáveis de saída. Elas então podem ser utilizadas quando se tem informações conhecidas e se deseja inferir informações desconhecidas (Patterson, 1996; Fausett, 1994). Alguns exemplos são:

Predição em Bolsa de Valores.

Sabe-se os preços das ações na semana passada e o índice atual; quer se saber quais serão os valores das ações no mercado amanhã.

Controle.

Quer se saber se o robô deve virar para esquerda, para a direita ou seguir em frente de modo a atingir um alvo; conhece-se o ambiente que a câmera do robô está observando no momento.

Um requisito importante para o emprego de uma rede neural é saber se existe uma relação entre as entradas propostas e as saídas desconhecidas. Contudo, esta relação pode ser ruidosa, ou seja, possuir outros fatores que não pertençam ao conjunto de entradas, que influenciam o comportamento das saídas ou até existir algum elemento de comportamento puramente aleatório.

De forma geral, quando se emprega uma rede neural não se sabe a equação entre as entradas e as saídas – se soubesse se modelaria diretamente. Outro fator importante nas redes neurais é que a mesma aprende a relação que existe entre as entradas e as saídas através de aprendizagem. Existem dois tipos de aprendizagem ou treinamento empregados em redes neurais, os supervisionados e não supervisionados, do qual treinamento supervisionado é o mais comum e será discutido na próxima seção.

Em aprendizado supervisionado o usuário da rede agrupa um conjunto de dados de treinamento. Os dados de treinamento contém exemplos de entradas com suas saídas correspondentes, e a rede aprende a inferir a relação entre as duas. Os dados de treinamento são coletados geralmente de registros históricos. Nos exemplos acima, pode-se incluir os valores de ações de mercado e os índices obtidos; registros anteriores de empréstimos bem sucedidos, incluindo-se questionários e um registro se o cliente foi anteriormente um bom pagador ou não; posições de robô padrão e suas reações corretas.

A rede neural é então treinada utilizando-se um dos algoritmos de aprendizagem supervisionada do qual o melhor exemplo é o *back propagation*, desenvolvido por (Rumelhart et. al., 1986), o que emprega os dados para ajustar os pesos da rede e suas polarizações de modo a minimizar os erros nas predições no conjunto de treinamento. Se a rede for treinada apropriadamente, aprendeu-se a modelar a função desconhecida que relaciona as variáveis de entrada às variáveis de saída, e pode subsequenteiramente realizar predições onde a saída for desconhecida.

3.5 . COLETANDO INFORMAÇÕES PARA A REDE NEURAL

As redes neurais processam dados numéricos em uma faixa limitada. Pode ser problemático se o dado for de valor fora da faixa usual de trabalho, se for um dado perdido ou um dado não numérico. Porém, existem métodos para se lidar com tais problemas. Dados numéricos são escalonados em uma faixa de valores apropriados para a rede e valores perdidos podem ser substituídos ao se empregar valores médios (ou outro tipo de estatística) da variável em relação aos casos de treinamento disponíveis (Bishop, 1995).

Manipular dados não numéricos é muito difícil. O tipo mais comum de dados não numéricos consiste de variáveis de valores nominais tais como gênero = {macho, fêmea}. Variáveis de valores

nominais podem ser representados numericamente. Contudo, as redes neurais não se comportam bem com variáveis nominais as quais possuem uma grande variedade de valores possíveis.

Por exemplo, considere uma rede sendo treinada para estimar o valor de propriedades residenciais. Os preços das casas dependem criticamente da área de uma cidade nas quais estão localizadas. Uma cidade em particular deve ser subdividida em dezenas de localizações e então se torna natural o emprego de uma variável nominal representando tais localizações. Infelizmente, pode ser difícil treinar uma rede neural diante de tais circunstâncias e a aproximação mais provável para se realizar tal tarefa é se criar pesos baseados em conhecimento de especialistas da área para cada localidade.

Outros tipos de dados não numéricos podem ser convertidos para a forma numérica ou descartados. Datas e tempos, se forem importantes, serão convertidas para valores ajustados a partir de uma data/tempo específico. Valores monetários são facilmente convertidos. Campos de texto tais como nomes podem não ser facilmente manipulado e deveriam ser descartados.

O tamanho da amostra para se treinar as redes neurais geram frequentemente dificuldades. Existem alguns recursos heurísticos que relacionam o número de valores necessários para o tamanho da rede, sendo o mais simples aquele em que a quantidade de valores deveria ser dez vezes os números das conexões da rede. Na verdade, o número necessário está também relacionado com a complexidade desconhecida da função embutida que cada rede precisa modelar. Conforme se aumenta o número de variáveis aumenta-se o número de valores de forma não linear, de modo que mesmo com um número pequeno de variáveis, um grande número de valores será empregado.

Para a maioria dos problemas práticos, o número de valores necessários será de centenas ou milhares. Para problemas mais complexos serão necessários mais valores, mas seria raro problemas com um número menor do que uma centena. Se seus dados forem mais esparsos do que isto, não possui informação suficiente para treinar a rede, a melhor solução seria ajustar um modelo linear.

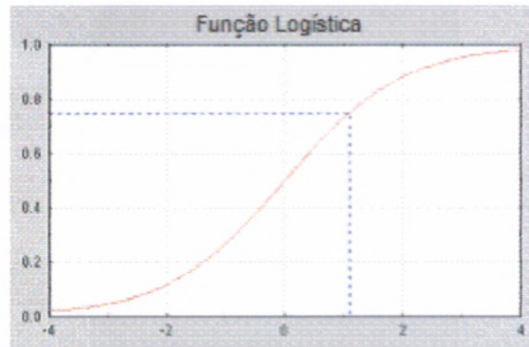
Muitos problemas práticos sofrem com dados não confiáveis: algumas variáveis podem se corromper com ruído ou valores perdidos. Apesar das redes neurais serem tolerantes a ruído, existe um limite para esta tolerância: se ocorrer o aparecimento de alguns *outliers* bem distante da faixa de valores normais poderá ocorrer um desvio no treinamento. A melhor solução seria identificar tais *outliers* e removê-los.

3.6 . PRÉ E PÓS PROCESSAMENTO

Todas as redes neurais pegam entradas numéricas e produzem saídas numéricas. A função de transferência de um neurônio é tipicamente escolhida de maneira a se aceitar entradas de qualquer escala e produzir saída em uma faixa estritamente limitada. Embora a entrada possa ser de qualquer

escala, ocorrerá o efeito de saturação de modo que a célula se tornará sensível apenas às entradas dentro de uma faixa limitada de valores. O gráfico abaixo mostra uma das funções de transferência mais comuns, a função logística também conhecida como função sigmoidal. Neste caso, a saída está na faixa de $(0,1)$ e a entrada será sensibilizada para uma faixa de valores entre $(-1,+1)$. A função é suave e facilmente diferenciável, fatores críticos para garantir que os algoritmos de treinamento trabalhem bem.

Figura 3.2 Função de ativação logística (sigmoidal)



Fonte: StatSoft's Electronic Book Homepage

Devido à resposta numérica de faixa limitada, além do fato das informações serem da forma numérica, isto implica que soluções neurais requerem estágios de pré-processamento e pós-processamento antes de serem empregados em aplicações reais (Bishop, 1995).

Escalonamento.

Os valores numéricos terão que ser escalonados dentro de uma faixa apropriada para a rede. Geralmente os dados de entrada se escalonam de forma linear e pode-se empregar algoritmos que calculam ponto de mínimo, máximo, média e desvio-padrão para se escalonar os valores numéricos dentro de uma faixa apropriada.

Em algumas circunstâncias o escalonamento deverá ser não linear caso seja apropriado, por exemplo, se verificar que a variável possui uma distribuição exponencial, seria apropriado utilizar uma escala logarítmica.

Valores Nominais.

Valores nominais podem ser dicotômicos (exemplo: gênero = {macho, fêmea}) ou de múltiplos estados (exemplo: mais do que dois estados). É fácil se transformar uma variável nominal dicotômica em valores numéricos (exemplo: Macho = 0, Fêmea = 1). Variáveis nominais

de múltiplos estados são mais difíceis de serem manipuladas. Poderiam ser representadas se empregando um valor numérico (exemplo: Cão = 0, Papagaio = 1, Gato = 2) mas isto implicaria em um ordenamento falso de valores nominais – neste caso, o papagaio estaria de certa forma no meio do caminho entre Cães e Gatos. Uma forma melhor de se representar seria a codificação no formato de *um de N* que é empregar um número de variáveis numéricas para representar a variável nominal simples. O número de variáveis numéricas iguala o número de valores possíveis; uma das N variáveis é fixada, enquanto as outras são zeradas (exemplo: Cão = {1,0,0}, Papagaio = {0,1,0}, Gato = {0,0,1}). Infelizmente, uma variável nominal com um grande número de estados poderá exigir um número proibitivo de variáveis numéricas de codificação *um de N*, aumentando o tamanho da rede neural e tornando difícil o seu treinamento. Em tais casos se torna mais viável empregar a representação através de um índice numérico simples.

Problemas de predição podem ser divididos em duas categorias principais: **classificação e regressão**.

Em classificação, o objetivo é determinar a que classe distinta certo dado de entrada pertence. Exemplos incluiriam aceitação de permissão de crédito bancário (sem risco, com risco), detecção de tumor canceroso (tumor, limpo), reconhecimento de assinatura (falso, verdadeiro). A forma mais comum de classificação se baseia em tarefas de dois estados, embora possam surgir tarefas com múltiplos estados de classificação.

Em regressão, o objetivo é prever o valor geralmente de uma variável contínua: valor de uma ação no mercado, o consumo de combustível de um carro, lucros para o próximo ano. Neste caso, a saída necessária será um valor numérico variável.

As redes neurais podem geralmente executar tarefas tanto de regressão ou classificação, embora seja mais comum executar cada tipo de tarefa distintamente. Na maioria dos casos, embora a rede possua uma variável de saída, no caso de problemas de classificação de valores de múltiplos estados, poderá existir diversos neurônios de saída.

3.7 . PERCEPTRONS MULTICAMADA (MLP)

Esta é talvez a arquitetura mais popular de redes em uso hoje em dia, proposta originalmente por Rumelhart e McClelland (1986) e discutida exaustivamente em livros sobre redes neurais (exemplo: Bishop, 1995). Este é o tipo de rede discutida brevemente em seções anteriores: as células executam uma soma ponderada com uma polarização e passa este nível de ativação para uma função de transferência que produz sua saída e as células são dispostas em uma topologia de camadas de transmissão contínua ou “*feedforward*”. Desta maneira a rede possui uma interpretação simples na forma de um modelo de entrada e saída com pesos e polarizações que

definem os parâmetros livres do modelo. Tais redes podem modelar funções de complexidade arbitrária e o número de células em cada camada, determinando a complexidade da função. Artigos importantes sobre desenvolvimento de redes MLP incluem especificações de número de camadas escondidas e o número de neurônios (células) nestas camadas. (veja Haykin, 2001; Bishop, 1995).

O número de entradas e saídas é definido pelo problema em particular, havendo uma certa incerteza sobre o número preciso de entradas a serem utilizado, um ponto a ser discutido posteriormente. Contudo, para o momento, assume-se que as variáveis de entrada são intuitivamente selecionadas e são todas significativas. O número de camadas escondidas e neurônios necessários toma-se como um bom ponto de partida se empregar uma camada escondida, com o número de neurônios igual à metade da soma do número de entradas com o número de saídas.

3.8 . TREINANDO PERCEPTRONS MULTICAMADA

Uma vez escolhida o número de camadas, o número de neurônios em cada camada, os pesos e as polarizações empregadas pela rede devem ser ajustados de modo a minimizar o erro de predição pela rede. Esta é a tarefa dos *algoritmos de treinamento*. Os valores históricos coletado automaticamente ajustarão os pesos e polarizações de maneira a minimizar o erro. Este processo equivale a ajustar o modelo representado pela rede aos dados de treinamento disponíveis. O erro de uma configuração em particular da rede poderá ser determinado por se empregar todos os dados de treinamento com a rede, comparando as saídas geradas com os valores de saída desejados. As diferenças são combinadas através de uma *função de erro* para definir o erro da rede. A função de erro mais comum é o somatório de erro quadrático onde os erros individuais de cada neurônio são elevados ao quadrado e somados.

Um conceito útil é o emprego da superfície de erro. Cada um dos N pesos e polarizações de uma rede, que são os parâmetros livres do modelo, são empregados como um espaço dimensional. A dimensão $N+1$ é o erro da rede. Para qualquer configuração possível dos pesos, o erro é representado graficamente em uma dimensão $N+1$, formando uma superfície de erro. O objetivo do treinamento da rede é encontrar o ponto de mínimo na superfície multidimensional.

Em um modelo linear usando a função de somatório de erro quadrático, a superfície de erro é uma parábola, o que significa que é uma parábola com um único ponto de mínimo global. Desta forma é fácil se encontrar o valor mínimo.

Mas as superfícies de erro das redes neurais são mais complexas, além de apresentarem valores indesejáveis conhecidos como mínimo local, que são os menores valores de um determinado trecho

da superfície, contudo ainda acima de um valor de mínimo global apresentam ainda pontos de sela, plateaus e ravinas estreitas.

Dependendo do método pode não ser possível analiticamente se determinar aonde se encontra um mínimo global de uma superfície de erro e, portanto, o treinamento da rede neural é essencialmente uma exploração da superfície de erro. A partir de uma configuração aleatória inicial de pesos e polarizações (exemplo: um ponto qualquer na superfície de erro, os algoritmos de erro buscam de forma incremental por um mínimo global. É feito tipicamente por se calcular o gradiente (rampa) da superfície de erro no ponto atual e então empregando esta informação se fazer um movimento do tipo *downhill* (montanha abaixo). Eventualmente, o algoritmo pára em um ponto inferior, o qual pode ser um mínimo local, mas que se espera tratar de um mínimo global.

3.9 . O ALGORITMO *BACK-PROPAGATION*

O algoritmo de treinamento mais conhecido é o *back-propagation* (veja Patterson, 1996; Haykin, 2001; Fausett, 1994). Os algoritmos modernos de segunda ordem tais como o *conjugate gradient descent* e o *Levenberg-Marquardt* (veja Bishop, 1995; Shepherd, 1997) são rápidos para muitos problemas, mas o *back-propagation* ainda possui vantagens em certas circunstâncias, além de ser o algoritmo mais fácil de se entender. Existem também modificações heurísticas do algoritmo *back-propagation* que trabalham bem com alguns problemas, tais como o *quick propagation* (Fahlman, 1988) e o *Delta-Bar-Delta* (Jacobs, 1988).

No *back-propagation*, o vetor de gradiente da superfície de erro é calculado. Este vetor aponta a linha de maior inclinação do ponto corrente, de modo que ao se mover ao longo desta distância, diminui-se o erro. Uma sequência de tais movimentos, desalecerando conforme se aproxima da parte inferior, o que eventualmente poderá ser um valor de mínimo. A dificuldade é sobre decidir sobre a magnitude dos passos a serem feitos.

Passos de grande magnitude podem realizar uma convergência mais rápida, mas pode também ultrapassar a solução se a superfície de erro for excêntrica, indo na direção errada. Um exemplo clássico disto em treinamento de rede neural é aonde o algoritmo progride lentamente através de uma descida, de um vale estreito, de um vale largo, balançando de um lado para o outro. Em contraste, passos curtos podem se direcionar para a direção correta, mas também requerem um grande número de iterações. Na prática, o tamanho do passo é proporcional à rampa, de modo que o algoritmo atinja um mínimo, para uma constante especial: a *taxa de aprendizagem*. O ajuste correto da taxa de aprendizagem depende da aplicação e é tipicamente escolhido por experimento; também pode ser variável com o tempo, se tornando menor conforme o algoritmo progride.

O algoritmo também é modificado pela inclusão de um termo de momento: que encoraja movimentos em uma direção fixa, de modo que se vários passos forem tomados na mesma direção, o algoritmo acelera, o que gera a habilidade de algumas vezes escapar de um mínimo local e também se mover rapidamente através de plateaus.

O algoritmo então progride iterativamente através de *épocas*. Em cada época, os valores de treinamento são submetidos à rede, e os valores alvos e as saídas atuais comparadas e os erros calculados. Este erro, junto com o gradiente da superfície de erro, é usado para se ajustar os pesos e, então, o processo se repete. A configuração inicial da rede é aleatória, e o treinamento pára quando um certo número de épocas é atingido, ou quando o erro alcança um nível aceitável, ou quando o erro pára de melhorar seus resultados. Pode se definir quais das condições de parada for melhor para se empregar.

3.10 . SOBRE APRENDIZADO E GENERALIZAÇÃO

O problema principal do aprendizado exposto é que não se minimiza realmente o erro em que estamos interessados – que é o erro da rede quando submetido por *novos* valores. Em outras palavras, a propriedade mais desejável de uma rede é sua habilidade em *generalizar* para novos casos. Na realidade, a rede é treinada para minimizar o erro com os valores de treinamento e como é difícil de se conseguir um conjunto grande de valores para treinamento, não é a mesma coisa que se minimizar o erro de uma superfície de erro “real”.

A manifestação mais importante desta distinção é o problema de aprendizado excessivo. É mais fácil se demonstrar este conceito empregando uma curva de ajuste polinomial do que uma rede neural, mas o conceito é precisamente o mesmo.

Um polinômio é uma equação com termos contendo somente constantes e grau de variáveis. Por exemplo:

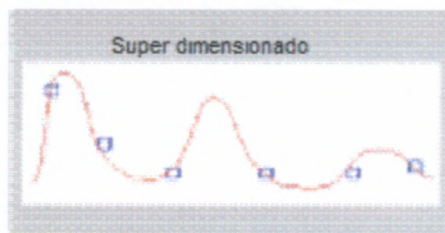
$$y = 2x + 3$$

$$y = 3x^2 + 4x + 1$$

Polinômios diferentes possuem formatos diferentes, com graus polinomiais maiores e portanto maior número de termos e possuem formas mais excêntricas. Dado um conjunto de dados, deseja-se ajustar uma curva polinomial para se explicar os dados. Os dados são geralmente ruidosos, portanto não se espera que o melhor modelo passe exatamente por todos os pontos. Um polinômio de ordem menor pode não ser suficientemente flexível para se ajustar a todos os pontos, aonde um polinômio

de ordem maior é mais flexível, se ajustando os dados exatamente por se adotar formas mais excêntricas que se relacionam com os pontos em questão.

Figura 3.3 Erro de generalização



Fonte: StatSoft's Electronic Book Homepage

As redes neurais têm o mesmo problema. Uma rede com um número maior de pesos modela uma função mais complexa. Uma rede com um número menor de pesos pode não ser suficiente para modelar uma função. Por exemplo, uma rede sem camada escondida modela apenas uma função linear.

Como se define a complexidade da rede? Uma rede maior invariavelmente atingirá um erro menor, mas pode resultar em aprendizado excessivo mais do que uma boa modelagem.

A resposta é utilizar uma verificação cruzada. Alguns dos dados de treinamento são reservados e não são empregados no algoritmo de *back-propagation*. No entanto, serão empregados para se fazer uma verificação independente no progresso do algoritmo. Conforme o treinamento progride, o erro de treinamento naturalmente decai se o treinamento converge e com o treinamento se minimiza a função de erro real, o erro de verificação cai também. Contudo, se o erro de verificação para de cair, ou começar a subir, isto indica que a rede está iniciando um processo de aprendizado excessivo dos dados e o treinamento deve ser paralisado. Quando isto ocorre, é aconselhável se diminuir o número de neurônios escondidos e/ou camadas escondidas, porque a rede está superdimensionada para o problema em questão. Em contraste, se a rede não for suficientemente potente para modelar a função em questão, o aprendizado excessivo não ocorre, e nem o erro de treinamento e nem o erro de verificação cai para um nível satisfatório.

Os problemas associados com o mínimo local e decisões sobre o tamanho da rede em uso, implica que usar uma rede neural tipicamente envolve experimentar com um grande número diferente de redes, provavelmente treinando cada uma um certo número de vezes para evitar ser enganado por um mínimo local, e se observando o comportamento individual. O guia chave para se verificar o comportamento da rede é o erro de verificação. Todavia, um modelo simples é sempre preferível em comparação a um modelo complexo, pode-se também selecionar uma rede menor em preferência a uma maior com uma melhoria insignificante no erro de verificação.

Sua confiabilidade como um guia independente para o comportamento do modelo será comprometido – com experimentos suficientes, deve-se esperar ter “sorte” de que a rede se comporte bem no conjunto de verificação. Para se adicionar confiança no comportamento final do modelo, é prática normal, ao menos onde o volume de dados de treinamento permitem, reservar um terço dos valores – o conjunto de teste. O modelo final é testado com os dados do conjunto de teste, para garantir que os resultados na verificação e no treinamento sejam reais, e não artefatos do processo de treinamento. É claro que para se concretizar esta tarefa apropriadamente, o conjunto de testes deve ser utilizado apenas uma vez – se for usado para ajustar e interagir com o processo de treinamento, se tornará efetivamente dados de verificação!

Resumindo-se, o desenvolvimento de uma rede, uma vez que as variáveis de entrada são selecionadas, segue-se os seguintes estágios:

- Selecione uma configuração inicial, tipicamente, com uma camada escondida de neurônios com metade da soma das entradas e saídas.
- Iterativamente, conduza um número de experimentos com cada configuração, retendo a melhor rede encontrada em termos de erro de verificação. Um número de experimentos são necessários com cada configuração para se evitar ser enganado quando se treina para localizar um mínimo local.
- Em cada experimento, se ocorrer subaprendizado, a rede não atinge um nível de comportamento aceitável, tente adicionar mais neurônios para a(s) camada(s) escondida(s). Se não for útil, tente adicionar uma camada escondida extra.
- Se sobre aprendizado ocorrer, quando o erro de verificação começar a subir, tente remover alguns neurônios escondidos, e possivelmente alguma camada.

3.11 . SELEÇÃO DE DADOS

Todos os estágios acima dependem da seguinte conclusão: especificamente, o treinamento, a verificação e os dados de teste devem ser representativos do modelo em questão e, os três conjuntos devem ter uma representatividade independente. Se os dados de treinamento não forem representativos, então o valor do modelo estará comprometido. Desde o início é importante se definir os tipos de problemas que podem corromper um conjunto de treinamento:

O Futuro Não é o Passado.

Os dados de treinamento são tipicamente históricos. Se as circunstâncias mudarem, relacionamentos que existiam no passado podem não persistir no futuro, e o modelo obtido a partir dos dados passados não será capaz de boas previsões.

Toda Eventualidade Deve Ser Coberta.

Uma rede neural só pode aprender casos que estão presentes no conjunto de dados, não se pode esperar que a rede faça uma decisão correta quando se deparar com um caso não previsto.

A Rede Aprende os Padrões Mais Fáceis.

Uma ilustração clássica, possivelmente apócrifa, é o do projeto de visão desenvolvida para automaticamente reconhecer tanques. Uma rede que é treinada com centenas de fotos incluindo tanques e uma centena sem tanques, alcança 100% de precisão. Mas quando testada com dados novos se apresenta ineficaz. A razão? As fotos com os tanques foram tiradas no escuro, em dias chuvosos; as fotos sem tanques em dias ensolarados. A rede aprende a distinguir o método trivial de diferenças em intensidade luminosa máxima. Para funcionar a rede deveria ser treinada incluindo-se todas as condições climáticas e de luz em que se espera que trabalhe – sem mencionar todos os tipos de terreno, ângulos de disparo, distâncias, etc.

Conjunto de Dados Não Balanceados.

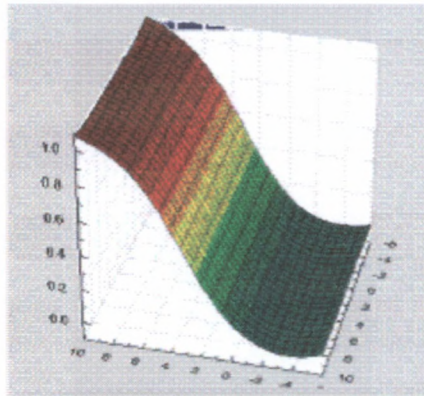
Desde que uma rede minimiza o erro sobre o conjunto de dados, as proporções dos tipos de dados de um conjunto são críticas. Uma rede treinada com 900 valores bons e 100 valores ruins será polarizada em suas decisões pelos casos bons, conforme o algoritmo diminui o seu erro geral, o qual será fortemente influenciado pelos casos bons. Se a representação de casos bons e ruins for diferente na população real, as decisões da rede serão equivocadas. Um bom exemplo seria o diagnóstico de doença. Talvez 90% dos pacientes rotineiramente testados estão livres da doença. Uma rede é treinada com dados disponíveis com uma proporção de 90/10. Então é empregada no diagnóstico em pacientes que reclamam de problemas específicos, onde a probabilidade de estar com a doença é de 50/50. A rede agirá com cautela e falhará no reconhecimento da doença em pacientes doentes. Em contraste, se for treinada com os dados dos reclamantes e então, testado nos dados de rotina, poderá aumentar um número significativo de falsos positivos. Em tais

circunstâncias, o conjunto de dados deve ser lapidado levando-se em conta a distribuição dos dados (exemplo: poderia se replicar um número menor de dados ou remover alguns dados numerosos), ou as decisões da rede devem ser modificadas pela inclusão de uma *matriz de perda* ou *loss matrix* (Bishop, 1995). Geralmente, a melhor solução é assegurar uma representação homogênea dos dados diversos, então se interpretar as decisões da rede de acordo.

3.12 . DETALHES SOBRE TREINAMENTO DE UMA MLP

Mais detalhes importantes sobre o comportamento de uma MLP e seu treinamento podem ser adquiridos por se considerar os tipos de funções que modelam. Relembrando que o nível de ativação de um neurônio é a soma ponderada das entradas mais um valor de polarização. Isto implica que o nível de ativação é uma função linear simples de entradas. A ativação é então repassada para uma curva Sigmoidal.

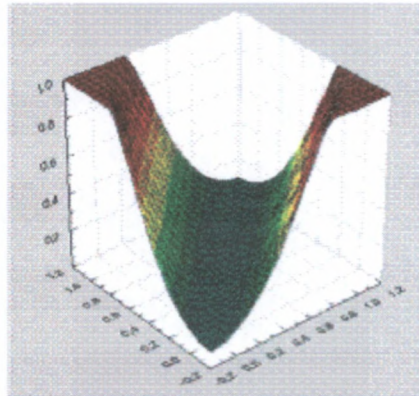
Figura 3.4 Superfície de resposta sigmoidal para duas entradas



Fonte: StatSoft's Electronic Book Homepage

A combinação da função linear multidimensional e da função Sigmoidal unidimensional nos dá a resposta característica de uma “queda sigmoidal” da primeira camada escondida de neurônios de uma MLP, o gráfico acima ilustra o formato obtido com duas entradas. Uma MLP com mais entradas terá uma versão de maior dimensão desta forma funcional. Alterando-se os pesos e as polarizações altera-se esta superfície de resposta. Em particular, tanto a orientação da superfície quanto o ângulo de declive podem ser alterados. Um declive acentuado corresponde a valores de peso maiores: dobrando-se todos os valores de peso tem-se a mesma orientação, mas um declive diferente.

Figura 3.5 Curva de resposta típica para uma rede com apenas uma camada de neurônios escondidos



Fonte: StatSoft's Electronic Book Homepage

Uma rede multicamada combina um número destas superfícies de resposta agrupadas, através de combinação linear repetida e funções de ativação não linear. O gráfico acima ilustra uma superfície de resposta típica para uma rede com apenas uma camada de neurônios escondidos, e um neurônio de saída, no problema clássico de OU exclusivo. Duas superfícies sigmoidais separadas foram combinadas no formato de uma superfície em "U".

Durante o treinamento da rede, os pesos e polarizações são iniciados com valores aleatórios menores. Isto implica que as superfícies estão cada uma alinhada aleatoriamente com um declive pequeno: eles estão efetivamente sem comprometimento com o modelo. Conforme o treinamento progride, as superfícies de resposta dos neurônios são rotacionadas e deslocadas em posições apropriadas, e as magnitudes dos pesos crescem conforme eles se comprometem para modelar partes particulares no modelamento da superfície de resposta alvo.

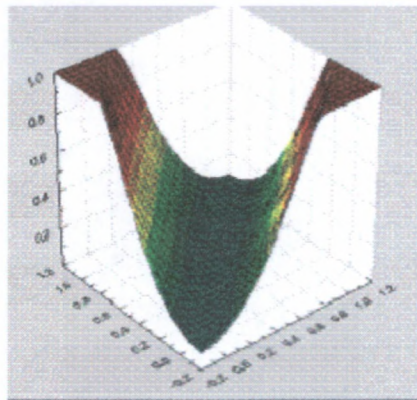
Em um problema de *classificação*, a tarefa do neurônio na saída é transmitir um sinal forte quando o caso pertencer a esta classe, e um sinal fraco se não pertencer. Em outras palavras, tentando modelar uma função que possui magnitude um para partes pertencentes ao espaço padrão que os contém e magnitude zero para os demais. Isto é conhecido como uma *função discriminante* em problemas de reconhecimento de padrões. Uma função discriminante ideal deveria ter uma estrutura do tipo plateau, onde todos os pontos da função estão tanto no nível zero quanto no nível um.

Se não houver neurônios escondidos, a saída só poderá modelar um "declive sigmoidal" com áreas de um lado em nível baixo e áreas do outro lado em nível alto. Sempre existirão regiões no meio (no declive) onde a altura é intermediária, mas conforme a magnitude dos pesos aumentam, esta área encolhe.

Esta curva caracteriza uma função discriminante linear. Pontos em um extremo do declive são classificados como pertencentes à classe, enquanto pontos no outro extremo não pertencem. Isto implica que uma rede sem camadas escondidas somente pode realizar separação de problemas lineares, aqueles em que uma linha ou, generalizando em dimensões maiores, um hiperplano, o qual separam os pontos no espaço padrão.

Uma rede com uma camada de neurônios escondida possui um número de declives sigmoidais (um para cada camada escondida) representado na camada escondida, e estas são combinadas em um plateau na camada de saída. O plateau tem um aspecto convexo. Embora o plateau seja convexo, pode se estender para o infinito em algumas direções. Tais redes podem na prática modelar adequadamente a maioria dos problemas de classificação do mundo real.

Figura 3.6 A resposta criada secciona o espaço ao longo de uma diagonal



Fonte: StatSoft's Electronic Book Homepage

Conforme apresentado acima, a superfície de resposta criado pela MLP para solucionar o problema do OU exclusivo: conforme pode se visualizar, secciona o espaço ao longo de uma diagonal.

Uma rede com duas camadas escondidas possui um número de plateaus combinados – o número de plateaus correspondem ao número de neurônios na segunda camada, e o número de lados de cada plateau corresponde ao número de neurônios na primeira camada escondida. Com um pouco de raciocínio pode se demonstrar que se pode representar qualquer formato, incluindo-se concavidades e buracos, empregando-se um grande número de plateaus.

Uma consequência destas observações é que uma MLP com duas camadas escondidas é teoricamente suficiente para modelar qualquer problema, existe uma prova mais formal, o Teorema de Kolmogorov. Isto não implica que uma rede com mais camadas não seja conveniente ou facilmente modelada para um problema em particular. Na prática, contudo, muitos problemas

tendem a considerar apenas uma camada escondida, com duas camadas alguns problemas ocasionais e com três praticamente desconhecido.

Uma questão chave em classificação é como interpretar os pontos "próximos" do declive. A prática padrão é se adotar alguns níveis de confiança, limites de aceitação e rejeição, os quais devem ser excedidos antes que o neurônio seja excitado para realizar uma decisão. Por exemplo, se limites de aceitação de 0.95 e de rejeição de 0.05 forem usados, uma saída com nível que exceda 0.95 será ativada, abaixo de 0.05 será zerada, e entre tais valores será classificada como indecisa.

Uma interpretação mais usual é se tratar as saídas como probabilidades. Neste caso, a rede dá mais informação do que simplesmente uma decisão: o qual habilitará que a saída da rede neural seja interpretada como probabilidades, o que significa que a rede aprende efetivamente a modelar a função de densidade de probabilidade da classe. Contudo, a interpretação probabilística somente é válida sob certas afirmações sobre a distribuição dos dados, especificamente, se for gerada pela família de distribuições exponenciais; veja Bishop, (1995). Uma decisão de classificação deve ainda ser feita, mas uma interpretação probabilística permite um conceito mais formal da tomada de decisão de "custo mínimo" envolvida.

3.13 . OUTROS ALGORITMOS DE TREINAMENTO DE UMA MLP

Anteriormente nesta seção, discutiu-se como o algoritmo de *back-propagation* realiza a descida de gradiente na superfície de erro. Falando mais claramente, calcula-se a direção do maior declive de descida na superfície, então se desloca na superfície uma distância proporcional à taxa de aprendizagem e ao declive, usando um momento quando se mantém uma direção consistente. Como analogia, se comporta como um canguru vendado pulando na direção mais óbvia. Atualmente, a descida é calculada independentemente na superfície de erro para cada valor de treinamento, e de forma aleatória, mas trata-se apenas de uma boa aproximação para a descida na superfície de erro. Outros algoritmos de treinamento para MLP trabalham de forma diferente, mas todos empregam uma estratégia desenvolvida para caminhar em direção a um mínimo o mais rápido possível.

Outras técnicas mais sofisticadas para funções não lineares de otimização estão se empregando. Entre estas podemos mencionar o *conjugate gradient descent* e o *Levenberg-Marquardt* (Bishop, 1995; Shepherd, 1997), os quais são formas bem sucedidas de dois tipos de algoritmo: busca em linha e região de confiança.

O algoritmo de busca em linha trabalha da seguinte forma: pega uma direção sensível para se movimentar na superfície multidimensional. Então projeta uma linha nesta direção, localiza o

mínimo nesta linha, sendo relativamente trivial se localizar um mínimo em uma linha por se empregar um algoritmo no formato de biseção, e repete. O que é uma "direção sensível" neste contexto? Uma escolha óbvia seria a direção da maior inclinação descendente, a mesma direção que foi escolhida pelo *back-propagation*. Na verdade, esta escolha intuitivamente óbvia prova-se ser um pouco pobre. Ao se minimizar em uma direção, a próxima linha descendente poderá "estragar" a minimização concernente à direção inicial, mesmo em uma superfície simples como uma parábola onde um grande número de buscas em linhas será necessário. Uma melhor opção é selecionar um conjugado ou direções que não interferem – por isso chamado *conjugate gradient descent* (Bishop, 1995).

A idéia aqui é que, uma vez que o algoritmo minimizou em uma direção particular, a segunda derivada naquela direção deveria se manter em zero. As direções conjugadas são selecionadas para manter esta segunda derivada em zero na presunção de que a superfície é parabólica, de outra maneira, "uma superfície bem lisa". Se esta condição persistir, as N épocas serão suficientes para se alcançar um mínimo. Na realidade, em uma superfície de erro complexa os conjugados se deterioram, mas o algoritmo ainda requer menos épocas do que o *back propagation*, e também converge para um mínimo melhor para se estabilizar, o *back propagation* deve trabalhar com uma taxa de aprendizagem extremamente baixa.

A região de confiança trabalha conforme a seguir: em vez de seguir uma direção de busca, assume-se que a superfície é de formato simples tal que o mínimo pode ser localizado e pular diretamente para ele – se a escolha for verdadeira. Tente o modelo e veja quão bom o ponto sugerido é. O modelo tipicamente assume que a superfície possui um formato bem comportado (por exemplo: uma parábola), o qual será verdadeiro se estiver suficientemente próximo de um mínimo. De outro forma, a escolha poderá ser grosseira e o modelo escolher pontos inapropriados para se movimentar. O modelo somente é confiável dentro de uma região do ponto corrente, e o tamanho da região é desconhecido. Portanto, escolha novos pontos para testar o comprometimento entre o valor sugerido pelo modelo e o sugerido pelo salto do gradiente padrão. Se o novo ponto for bom, move-se para lá, e fortalece a atuação do modelo em selecionar um novo ponto; se for ruim, não mova, e fortaleça o passo do algoritmo de *gradient descent* (descida de gradiente) para selecionar um novo ponto e fazê-lo com passos menores. O *Levenberg-Marquardt* utiliza um modelo que assume que a função é localmente linear, e portanto possui uma superfície de erro parabólica.

O *Levenberg-Marquardt* (Levenberg, 1944; Marquardt, 1963; Bishop, 1995) é tipicamente o algoritmo de treinamento mais rápido, embora possua algumas limitações importantes, especificamente: só pode ser empregado em redes com uma única saída, somente pode ser utilizado em conjunto com a função de soma de erro quadrático e precisa uma quantidade de memória

proporcional a W^2 (onde W é o número de pesos de uma rede; tornando-se impraticável para redes grandes). O *Conjugate gradient descent* é bom e não sofre com tantas restrições.

O algoritmo *back propagation* ainda pode ser útil, não que venha a garantir uma solução rápida e precisa. Ainda é uma boa escolha se o conjunto de dados é relativamente grande e possuem bastantes dados redundantes. Os ajustes de erro caso a caso do algoritmo *back propagation* significa que a redundância de dados não causa prejuízo, por exemplo, se dobrar o tamanho do conjunto de dados ao se replicar cada caso, cada época levará duas vezes mais o tempo convencional, mas terá o mesmo efeito de duas épocas antigas, portanto não há perdas. Em contraste, o algoritmo *Levenberg-Marquardt* realiza cálculos empregando o conjunto inteiro de dados, aumentando-se o número de casos poderá desacelerar cada época significativamente, contudo não melhora necessariamente o comportamento desta época, fora casos em que os dados são redundantes; se os dados são esparsos, então a adição de dados fará com que cada época melhore. O *back propagation* é igualmente bom se o conjunto de dados for muito pequeno, porque existe informação insuficiente para se realizar uma solução refinada e apropriada, outro algoritmo mais avançado pode alcançar um erro de treinamento menor, mas o erro de verificação poderá não melhorar no mesmo modo.

Existem também duas variações do *back propagation* (o *quick propagation*, Fahlman, 1988 e o *Delta-bar-Delta*, Jacobs, 1988) que foram desenvolvidas para lidar com algumas das limitações desta técnica. Em muitos casos, não são significativamente melhores do que o *back propagation*, e algumas vezes elas são piores, com comportamento relativo e dependente da aplicação. Estes dois algoritmos também requerem mais parâmetros do que os demais já expostos anteriormente.

3.14 . CLASSIFICAÇÃO EM REDES NEURAIS

Em problemas de classificação, o propósito da rede é designar cada valor para uma das classes, ou de forma mais generalizada, estimar a probabilidade de ser membro de uma determinada classe.

Os valores nominais são normalmente representados em redes se empregando uma das duas técnicas, a primeira disponível apenas para valores dicotômicos; estas técnicas são: estado binário, *1 de N*. Em uma representação binária, um único nó corresponde à variável e um valor de 0 é interpretado como um dos estados e um valor de 1 para o outro. Em codificação de *1 de N*, um neurônio é alocado para cada estado, com a representação do nível 1 em um dos neurônios e o 0 em outro.

As variáveis de entrada são facilmente convertidas se empregando um dos métodos expostos acima, tanto durante o treinamento quanto durante a execução. As saídas alvo para os neurônios

correspondente às variáveis nominais são facilmente determinadas durante o treinamento. Contudo, necessita-se mais esforço para se determinar a classe de saída designada pela rede durante a execução.

Os neurônios de saída contêm valores contínuos entre 0 e 1. De maneira a definir definitivamente uma classe para as saídas, a rede deve decidir se as saídas se encontram relativamente próximas a 0 e 1. Se não estiverem, a classe é definida como “indecisa”.

As redes neurais empregam um nível de confiança, com limiares de aceitação e rejeição, para decidir como se interpretar as saídas de rede. Estes limiares podem ser ajustados para fazer com que a rede seja mais ou menos sensível na designação de classificação. A interpretação difere de forma indelével entre a representação dicotômica e a de *1 de N*:

Dicotômica.

Se a saída do neurônio estiver acima do limiar de aceitação, escolhe-se a classe do tipo 1. Se a saída estiver abaixo do limiar de rejeição, a classe 0 é designada. Se a saída estiver entre os dois limiares, a classe é definida como indecisa.

Um de N.

Uma classe é selecionada se a saída correspondente estiver acima do limiar de aceitação E todas as saídas dos outros neurônios de saída estiverem abaixo do nível de rejeição. Se esta condição não for satisfeita, a classe é designada como indecisa.

Existe uma peculiaridade quando se lida com codificação *1 de N*. Em primeira análise, pode-se esperar que o melhor valor de limiar de aceitação e rejeição seja de 0,5. Na verdade, no caso de *1 de N* isto não ocorre e sim no caso de rede dicotômica. Geralmente pode se ajustar o limiar de aceitação abaixo do de rejeição e a rede passa a ter, por exemplo, limiar de aceitação de 0 e de rejeição de 1. Este ajuste paradoxal implica em que o neurônio com o valor de ativação maior indica o tipo de classe, independente do valor dos demais (neurônios competitivos). Isto ocorre porque o algoritmo de designação de uma classe atua da seguinte forma:

- Selecione o neurônio com a maior saída. Se o neurônio possuir uma saída maior ou igual ao limiar de aceitação, e todas as demais saídas possuírem valores menores do que o limiar de rejeição, designe a classe representada por esta unidade.

Com um limiar de aceitação de 0, o neurônio campeão será certamente aceito e com um limiar de rejeição de 1, nenhuma das saídas dos demais neurônios serão desconsideradas, de modo que o algoritmo se reduz a simplesmente eleger como valor de saída a do neurônio que ganhou com o

maior valor. Em contraste, se ambos os limiares forem ajustados para 0,5, a rede pode retornar como resposta uma indecisão, se o nível de ativação do neurônio ganhador estiver abaixo de 0,5 ou qualquer um dos demais estiverem com o valor acima de 0,5.

Embora tais conceitos dependam um pouco de se fazer experimentos com estes limiares, os mesmos permitem que se estabeleçam algumas condições importantes. Por exemplo, se escolhermos um nível de aceitação de 0,3 e de rejeição de 0,7 conclui-se que: “selecione a classe com o neurônio vitorioso no caso de seu nível de saída for pelo menos de 0,3 e que nenhuma das outras saídas possuam valores de ativação maior do que 0,7”, em outras palavras, o ganhador deverá apresentar um certo nível de ativação significativo enquanto os perdedores não devem interferir para se atingir uma decisão.

Na discussão posterior se assume que uma classificação “positiva” é indicada por um valor próximo de 1 e uma classificação “negativa” por um valor próximo de 0. Isto se torna verdadeiro se empregarmos a função de ativação de saída logística e se torna conveniente com uma faixa de probabilidades excursionando entre 0 e 1. Contudo, em algumas circunstâncias pode ser conveniente se empregar outras faixas de ajuste. Também, algumas vezes a ordem pode ser reversa, com valores de saídas menores indicando positividade.

Primeiro, a faixa dos valores correspondem ao valor mínimo de média e ao valor máximo de desvio padrão armazenados para cada variável. Com uma função de ativação de saída logística os valores padrões de 0 e 1 são excelentes. Alguns autores recomendam o emprego de uma função de ativação do tipo tangente hiperbólica, o qual possui uma faixa de sinal de -1 a +1. O desempenho do treinamento é melhorado porque esta função, diferente da função logística, é simétrica. Desta maneira, ao se alterar os valores de média mínima e de desvio padrão máximo a rede automaticamente irá interpretar as classes corretamente.

3.15 . FOLHA DE DADOS DE CLASSIFICAÇÃO

Quando os limiares de aceitação / rejeição forem selecionados e se acertar a habilidade de classificação da rede, o indicador mais importante é a Folha de Dados de Classificação. Ela mostra quantos valores foram corretamente classificados, incorretamente classificados ou sem classificação. Também revela os valores que pertenciam a uma determinada classe e foram designadas para outra. Todos estes valores são levantados independentemente para os conjuntos de treinamento, validação e teste.

3.16 . PREDIÇÃO EM SÉRIES TEMPORAIS EM REDES NEURAI

Em problemas de séries temporais o objetivo é conseguir prever os valores seguintes de uma variável que se modifica com o tempo, empregando-se valores passados do mesmo e/ou outras variáveis (Bishop, 1995).

Tipicamente as variáveis preditas são contínuas, de modo que as séries temporais são um tipo especial de regressão.

É muito usual se prever os próximos valores de uma série temporal a partir de um número fixo de valores anteriores, observando-se um lapso de tempo à frente. Quando o próximo valor em uma série é gerado, valores futuros são estimados por se alimentar a rede com os novos valores junto com valores anteriores, o que chamamos de projeção de série temporal.

Em redes neurais pode se empregar qualquer tipo de rede para predição em uma série temporal, contudo, o tipo de rede deve ser compatível com o tipo de problema, se é de regressão ou classificação. A rede é configurada para a predição de uma série temporal se ajustando os parâmetros de passo e visualização prévia. O parâmetro de passo indica o número de valores de entrada que se deve alimentar e o parâmetro de visualização prévia quão distante a predição deve ser feita. A rede pode ter qualquer número de variáveis de entrada e saída. Contudo, muito comum é a utilização de uma variável única o qual é tanto entrada, levando-se em conta a visualização prévia, como saída. Configurar-se uma rede para utilização em série temporal altera a maneira de como os dados são pré-processados, por exemplo, é desenvolvido a partir de um número de valores sequenciais, mais do que um único caso, mas a execução da rede e o treinamento procede da mesma forma que qualquer outro problema.

O conjunto de treinamento para série temporal, portanto, possui uma única variável e é do tipo entrada/saída, por exemplo, que é empregada tanto para a entrada como para a saída da rede.

O conceito mais difícil em redes neurais empregando séries temporais de se lidar é a interpretação do treinamento, verificação, teste e valores ignorados. Para conjunto de dados padrões, cada valor é independente, e a compreensão destas informações fica clara. Contudo, em uma rede com série temporal cada padrão de entradas e saídas é na realidade gerado a partir de um número de casos, determinado pelos parâmetros de passo e visualização prévia. Existem duas consequências disto:

O tipo do padrão é determinado pelo tipo de informação de saída. Por exemplo, em um conjunto de dados contendo alguns casos, os dois primeiros são ignorados e o terceiro testado, com

passo = 2 e visualização prévia = 1, o primeiro padrão empregado é do tipo teste e, gera suas entradas a partir dos dois primeiros casos e sua saída do terceiro. Então, os dois primeiros valores são utilizados no conjunto de teste embora tenham sido marcados como ignorados. Além do mais, cada valor dado pode ser usado para três tipos de padrão, e estes podem ser qualquer um entre treinamento, verificação e teste. Em certo sentido, os dados se “diluem entre” treinamento, verificação e teste. Para se isolar os três conjuntos integralmente, se constrói blocos contíguos de treinamento, verificação e teste, separados por um número apropriado de valores ignorados.

Os primeiros poucos valores somente podem ser utilizados como entradas para padrões. Quando se seleciona dados para uso em uma série temporal, o número do valor selecionado é sempre um valor de saída. Os primeiros poucos valores não podem ser selecionados, já que necessitaria de valores adiantados antes do começo do conjunto de dados, e eles são automaticamente marcados como ignorados.

3.17 VANTAGENS DE SE EMPREGAR UMA REDE NEURAL RECORRENTE

As redes neurais de transmissão contínua ou *feedforward* são empregadas com sucesso para resolver problemas que computam funções estáticas, por exemplo, funções cuja saída depende do valor da variável de entrada atual e que ignora valores anteriores. Contudo, no mundo real e nesta aplicação específica que se está propondo costuma se deparar com situações que não podem ser resolvidas com funções estáticas porque a função calculada se modifica a cada entrada de variável recebida.

Um exemplo usual deste tipo de arquitetura de rede é a construção de um autômato determinístico finito que necessita de memorização de valores de entrada anteriores para definir o resultado de sua saída em relação ao valor corrente, que pode gerar valores distintos de saída para o mesmo valor corrente dependendo dos valores passados acumulados.

É importante salientar que a arquitetura *feedforward* em que os valores de entrada anteriores não influenciam o processamento de entradas futuras pode sofrer modificações para processar valores passados. Introduzindo-se conexões de realimentação na rede far-se-á com que os valores de entrada passados retornem para a camada de entrada e afetem o processamento de entradas futuras.

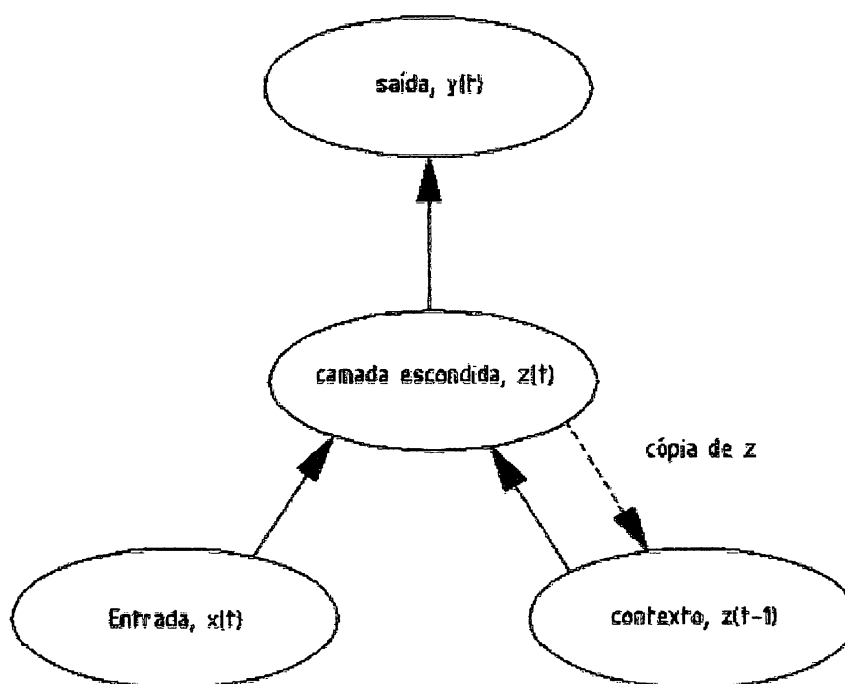
Esta classe de redes neurais que contém conexões de realimentação são chamadas de Redes Neurais Recorrentes (ou em inglês *Recurrent Neural Networks* – *RNN*). A topologia deste tipo de rede é bastante flexível, qualquer nó pode ser realimentado para qualquer outro e inclusive para si próprio. Os únicos requisitos rígidos para este tipo de rede são as entradas e as saídas que devem ser bem definidas.

Alguns problemas solucionados com redes recorrentes que se pode citar são: controle de articulações de robôs (Hornby et al., 2001); inferência gramatical em linguagem (Elman, 1990), identificação de presença de uma pessoa em uma determinada região vigiada por câmeras de segurança através de uma rede de Elman treinada para identificar características de movimentação biológica, como: o caminhar, correr ou saltitar (Giese, 2000); controle preditivo não linear de sistema de nível (Paulo Gil et al., 1999)

3.18 . ARQUITETURA DE UMA REDE DE ELMAN

A arquitetura da rede de Elman ou também denominada Rede Recorrente Simples (Elman, 1990) é representada na figura a seguir. Cada elipse na figura representa um conjunto de pontos de conexão. As conexões recorrentes nesta rede (em linha pontilhada) são processadas da seguinte forma: a cada intervalo de tempo t os neurônios da camada escondida recebem dados dos nós de entrada, os valores computados pelos neurônios são memorizados na camada de contexto (a elipse na parte inferior direita), cada unidade de contexto está conectada aos neurônios da camada escondida. A presença desta realimentação implica que as saídas dos neurônios no tempo t podem influenciar a resposta dos neurônios ocultos no tempo $t+1$. As unidades de contexto são conectadas com os neurônios da camada oculta num relacionamento de um para todos.

Figura 3.7 Representação esquemática de uma rede de Elman



4 . MATERIAL E MÉTODO

4.1 . COLETA DE DADOS

Os dados consistem em quatro variáveis climáticas fornecidas pela estação meteorológica nº 02651043 pertencente ao IAPAR (Instituto Agrícola do Paraná) situado no município de Palmas, P.R., na região sul do estado, a uma latitude de 26,216° sul e longitude de 51,083° oeste, altitude de 1035 m.

As quatro variáveis consistem em temperatura mínima no dia (° C), temperatura máxima no dia (° C), umidade relativa do ar (%) e velocidade do vento (m/s).

4.1.1 . Filtragem Dos Dados

Os dados fornecidos passaram por uma filtragem, porque houve períodos em que a estação teve problemas com o sistema de sensoriamento. Além disso, somente nos últimos 4 anos, (de Abril de 1998 até Dezembro de 2001) é que se passou a realizar a medição de velocidade do vento.

A matriz de pontos obtida após a filtragem ficou com a dimensão de 4 variáveis com 1249 observações.

4.1.2 . Teste De Normalidade Dos Dados

Para realizar o teste de normalidade dos dados, utilizamos o teste com a distribuição qui-quadrado, procedendo da seguinte maneira:

a) Cálculo da média de $p = 4$ variáveis e $n = 1249$ observações.

$$X_m = [11,7931 \ 18,3656 \ 77,0107 \ 2,6624]$$

Portanto obtêm-se os seguintes valores médios a partir da matriz acima:

$$X_{m1} = 11,7931 \text{ (valor médio da temperatura mínima)}$$

$$X_{m2} = 18,3656 \text{ (valor médio da temperatura máxima)}$$

$$X_{m3} = 77,0107 \text{ (valor médio da umidade relativa)}$$

$$X_{m4} = 2,6624 \text{ (valor médio da velocidade do vento)}$$

b) Se calcula a matriz de covariância:

$$S = \begin{bmatrix} 21,281 & 19,036 & 4,0356 & -0,19468 \\ 19,036 & 39,587 & -0,049973 & -0,278 \\ 4,0356 & -0,049973 & 59,252 & -0,2223 \\ -0,19468 & -0,278 & -0,2223 & 0,44555 \end{bmatrix}$$

c) Se calcula as distâncias generalizadas baseado em:

$$d_j^2 = (x_j - \bar{x})^t \cdot S^{-1} \cdot (x_j - \bar{x}) \quad (4.1)$$

d) Se ordenam estas distâncias em ordem crescente.

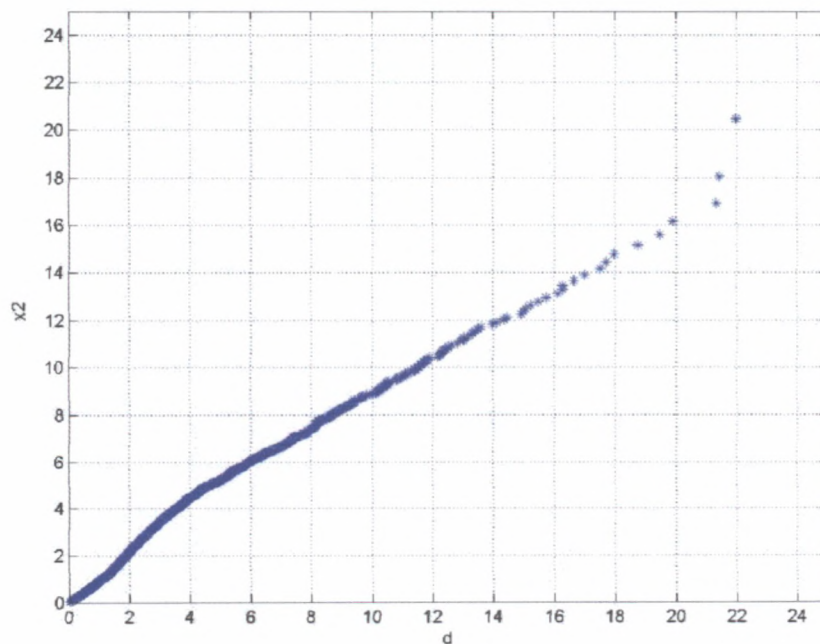
e) Se definem os percentis para a qui-quadrado.

f) Se obtém a inversa da qui-quadrado.

Observação: não será tabulado as variáveis por questão de sigilo.

Plotando-se o gráfico verifica-se a normalidade porque **os pontos situam-se nas proximidades de uma reta, não havendo desvio sistemático.**

Figura 4.1 Gráfico de normalidade dos dados



4.1.3 . Normalização Das Variáveis (Z1, Z2,Z3 E Z4)

De acordo com a teoria no capítulo 4.5 (Heurísticas para melhorar o desempenho do algoritmo de retropropagação), recomenda-se no item 5 a “normalização das variáveis de entrada” para acelerar aprendizagem dos pesos sinápticos para que venham a convergir rapidamente para o valor-alvo dentro do erro definido. Esta normalização impede que ocorra a saturação das saídas dos neurônios ocultos em sua função de ativação decorrentes de variáveis que possuem diferentes faixas de trabalho e que não correspondem ao valor típico destas funções (0,1) e, como o processamento das sinapses ocorre simultaneamente em paralelo, isto também agrava a ocorrência de saturação e não convergência do algoritmo de aprendizagem.

As recomendações para a normalização (LeCun, 1993) das entradas é que o conteúdo de treinamento tenha valor médio em zero, que as variáveis sejam descorrelacionadas e que o desvio-padrão de todas as variáveis empregadas seja unitário.

Nesta fase, aplicaremos apenas duas das três recomendações expostas anteriormente, a saber, média em zero e desvio-padrão unitário que é comumente conhecido como padronização de variáveis e facilmente aplicado pela seguinte fórmula:

$$z = \frac{(x - \bar{x})}{\sigma}$$

onde: (4.2)

$\bar{x}$ = valor médio

σ = desvio - padrão

1º Passo: obtenção do valor médio das quatro variáveis, que já foi calculado para o teste de normalidade dos dados.

$$X_m = [11,7931 \ 18,3656 \ 77,0107 \ 2,6624]$$

2º Passo: aproveita-se também o cálculo da matriz de covariância do teste de normalidade. A diagonal desta matriz corresponde ao valor das variâncias das quatro variáveis que se pretende normalizar, portanto, se extrair a raiz quadrada da diagonal desta matriz obtém-se um vetor de desvio-padrão:

$$s = \begin{bmatrix} 21,281 & 19,036 & 4,0356 & -0,19468 \\ 19,036 & 39,587 & -0,049973 & -0,278 \\ 4,0356 & -0,049973 & 59,252 & -0,2223 \\ -0,19468 & -0,278 & -0,2223 & 0,44555 \end{bmatrix}$$

E o vetor resultante é:

$$\sigma = \begin{bmatrix} 4,6131 \\ 6,2918 \\ 7,6975 \\ 0,6675 \end{bmatrix}$$

Obtêm-se, a partir deste vetor, os seguintes valores de desvio-padrão:

$\sigma_1 = 4,6131$ (desvio - padrão da temperatura mínima)

$\sigma_2 = 6,2918$ (desvio - padrão da temperatura máxima)

$\sigma_3 = 7,6975$ (desvio - padrão da umidade relativa)

$\sigma_4 = 0,6675$ (desvio - padrão da velocidade do vento)

3º Passo: com os valores médios e os valores de desvio-padrão se efetua a padronização das quatro variáveis para que tenha valor médio igual a zero e desvio-padrão unitário aplicando-se a fórmula (4.2). Não serão tabulados os valores porque compõe uma matriz 4 x 1249.

4.1.4 . Verificando A Correlação Entre As Variáveis

Agora se efetua uma verificação de correlação entre as variáveis para se determinar à influência entre as mesmas.

O coeficiente de correlação de Pearson mede a associação linear entre duas variáveis e não depende das unidades de engenharia de cada variável. O coeficiente de correlação para as variáveis i e j são definidos como:

$$r_{ij} = \frac{s_{ij}}{\sqrt{s_{ii}} \cdot \sqrt{s_{jj}}} = \frac{\sum_{k=1}^n (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{\sqrt{\sum_{k=1}^n (x_{ki} - \bar{x}_i)^2} \cdot \sqrt{\sum_{k=1}^n (x_{kj} - \bar{x}_j)^2}} \text{ para } i = 1, 2, \dots, n \text{ e } j = 1, 2, \dots, n \quad (4.3)$$

A matriz de correlação obtida com a definição acima é:

$$R = \begin{bmatrix} 1 & r_{12} & r_{13} & r_{1n} \\ r_{21} & 1 & r_{23} & r_{2n} \\ r_{31} & r_{32} & 1 & r_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{n1} & r_{n2} & r_{n3} & 1 \end{bmatrix} \quad (4.4)$$

A matriz de correlação R consiste de n linhas e n colunas, portanto uma matriz quadrada e, que também é simétrica, ou seja, $r_{ij} = r_{ji}$.

O coeficiente de correlação da amostra é uma versão padronizada da covariância da amostra, onde o produto das raízes quadradas das variâncias da amostra cria a padronização. Embora os sinais da correlação e da covariância sejam os mesmos, a correlação se torna mais fácil de se interpretar porque a sua magnitude é limitada. Fazendo-se um sumário, o coeficiente de correlação r possui as seguintes propriedades:

1. O valor de r varia de -1 a +1.

2. O valor de r mede a força da associação linear. Se $r = 0$, isto implica uma falta de associação linear entre as variáveis. Por outro lado, o sinal de r indica a direção da associação: $r < 0$ implica uma tendência de um valor do par ser maior em sua média enquanto o outro é menor em sua média; e $r > 0$ implica uma tendência para um valor do par aumentar ao mesmo tempo que o outro valor aumenta e também ambos diminuirão seus valores simultaneamente.

Com isto, obtém-se a matriz:

$$R = \begin{bmatrix} 1 & 0,6559 & 0,1136 & -0,0632 \\ & 1 & -0,0010 & -0,0662 \\ & & 1 & -0,0433 \\ & & & 1 \end{bmatrix} \quad (4.5)$$

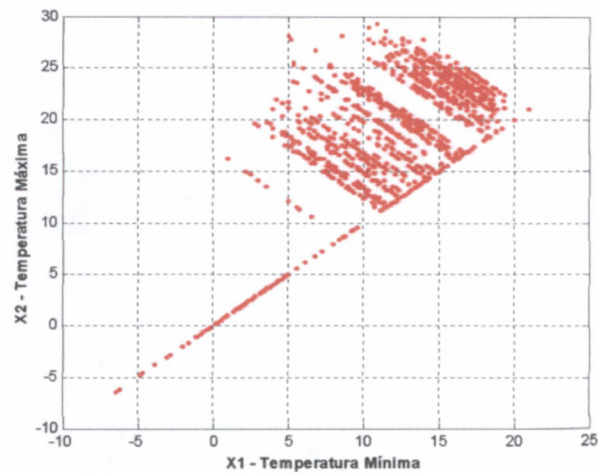
Na interpretação dos valores da matriz R temos:

4.1.4.1 . Correlação entre temperatura mínima e temperatura máxima (x_1 e x_2):

$r_{12} = r_{21} = 0,6559$, o que indica uma correlação positiva média entre a temperatura mínima do ar e a temperatura máxima do ar. (4.5.1)

Observando o gráfico de dispersão entre as variáveis x_1 (temperatura mínima) e x_2 (temperatura máxima) nota-se a tendência de se formar um alinhamento dos pontos.

Figura 4.2 Gráfico de dispersão de x_1 e x_2

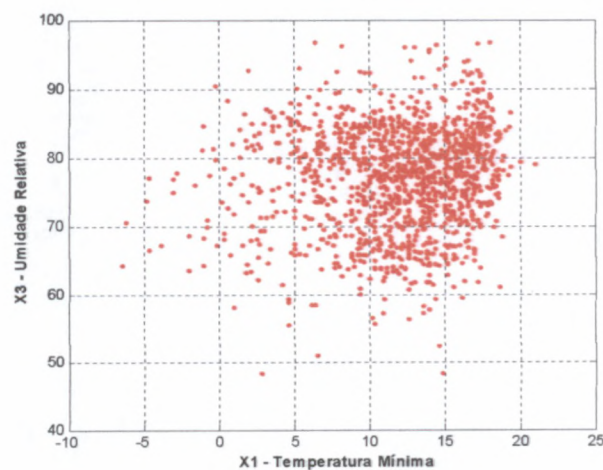


4.1.4.2 . Correlação entre temperatura mínima e umidade relativa (x_1 e x_3):

$x_{13} = x_{31} = 0,1136$, o que indica uma correlação positiva fraca entre a temperatura mínima do ar e a umidade relativa do ar. (4.5.2)

Observando o gráfico de dispersão entre as variáveis x_1 (temperatura mínima) e x_3 (umidade relativa) nota-se uma maior dispersão dos pontos.

Figura 4.3 Gráfico de dispersão de x_1 e x_3

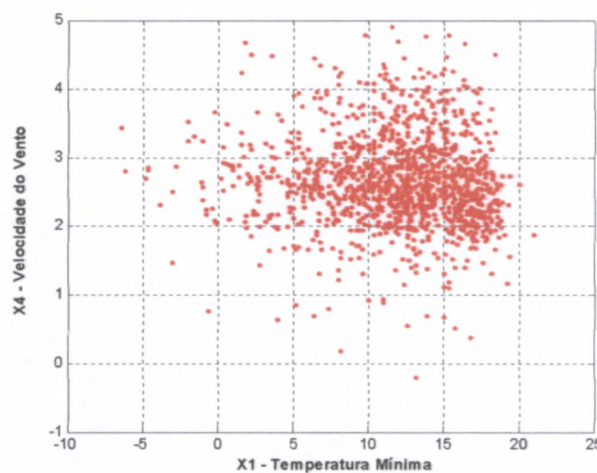


4.1.4.3 . Correlação entre temperatura mínima e velocidade do vento (x_1 e x_4):

$x_{14} = x_{41} = -0,0632$, o que indica uma correlação negativa muito fraca entre a temperatura mínima do ar e a velocidade do vento. (4.5.3)

Observando o gráfico de dispersão entre as variáveis x_1 (temperatura mínima) e x_4 (velocidade) nota-se uma maior dispersão dos pontos.

Figura 4.4 Gráfico de dispersão de x_1 e x_4

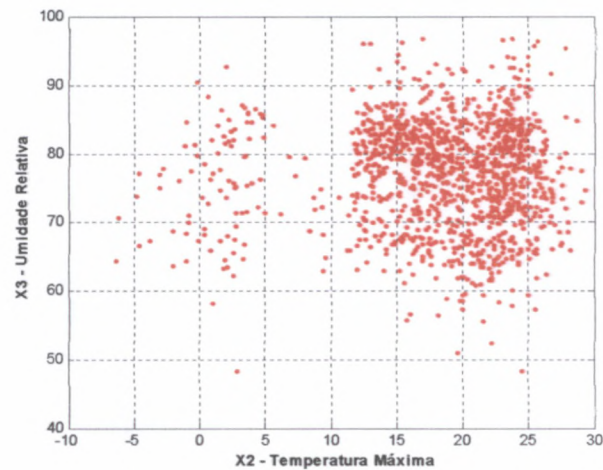


4.1.4.4 . Correlação entre temperatura máxima e umidade relativa (x_2 e x_3):

$x_{23} = x_{32} = -0,0010$, o que indica uma correlação nula entre a temperatura máxima do ar e a umidade relativa do ar. (4.5.4)

Observando o gráfico de dispersão entre as variáveis x_2 (temperatura máxima) e x_3 (umidade relativa) nota-se uma maior dispersão dos pontos em dois clusters.

Figura 4.5 Gráfico de dispersão de x_2 e x_3

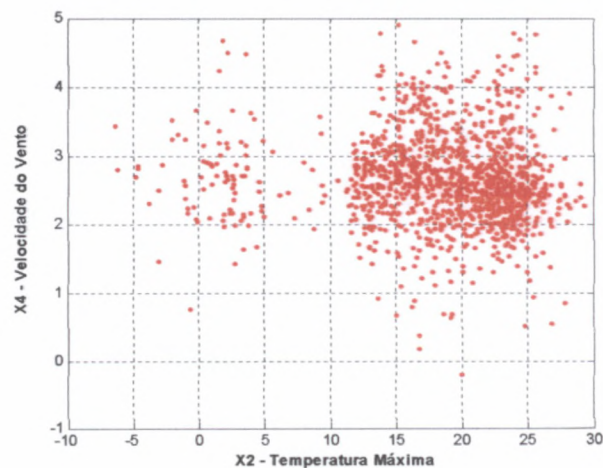


4.1.4.5 . Correlação entre temperatura máxima e velocidade do vento (x_2 e x_4):

$x_{24} = x_{42} = -0.0662$, o que indica uma correlação negativa muito fraca entre a temperatura máxima do ar e a velocidade do vento. (4.5.5)

Observando o gráfico de dispersão entre as variáveis x_2 (temperatura máxima) e x_4 (velocidade) nota-se uma maior dispersão dos pontos.

Figura 4.6 Gráfico de dispersão de x_2 e x_4

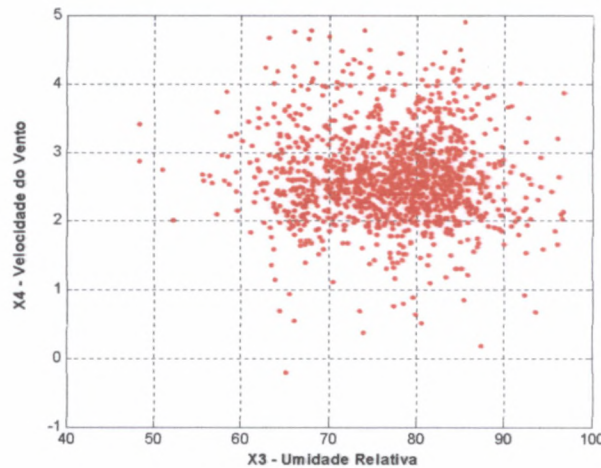


4.1.4.6 . Correlação entre umidade relativa e velocidade do vento (x_3 e x_4):

$x_{34} = x_{43} = -0.0433$, o que indica uma correlação negativa muito fraca entre a umidade relativa do ar e a velocidade do vento. (4.5.6)

Observando o gráfico de dispersão entre as variáveis x_3 (umidade relativa) e x_4 (velocidade) nota-se uma maior dispersão dos pontos.

Figura 4.7 Gráfico de dispersão de x_3 e x_4



4.1.4.7 . Conclusões sobre as correlações

Entre as variáveis temperatura mínima e temperatura máxima (x_1 e x_2) encontra-se o pior caso de correlação que pode interferir no desempenho de aprendizagem, já que o aumento ou decréscimo do valor dos pesos sinápticos de cada variável irá influenciar indiretamente na resposta da outra. É digno de nota se declarar que esta correlação positiva seria óbvia, já que as duas variáveis são adquiridas por se mensurar o valor de limite mínimo e máximo de uma mesma variável, a saber, temperatura em um intervalo de tempo fixo de 24 horas.

Na prática, a aplicação de um algoritmo com rede neural que se aplica estas duas variáveis esbarraria num problema técnico, precisaria de um período de 24 horas para registrar os valores de temperatura mínima e máxima para poder atualizar os valores de entrada para estas variáveis.

Outro problema seria o risco de se aguardar uma nova informação de temperatura mínima, o que geralmente ocorre de madrugada, para a rede tomar uma nova decisão e incorrer na possibilidade de formação da geada no mesmo momento ou anteriormente ao fechamento deste valor para a tomada de ação do algoritmo.

Por isso, em vez de se adotar a desconexão entre estas duas variáveis, é preferível se realizar um teste de eliminação de uma das variáveis e ver o comprometimento da resposta da rede sem esta informação. Este teste será o próximo passo de execução.

Entre as variáveis temperatura mínima e umidade relativa (x_1 e x_3) existe também uma correlação positiva, só que fraca, podendo se desprezar a desconexão. Também é interessante perceber que, levando-se em conta fenômenos termodinâmicos, realmente a diminuição da temperatura mínima pode acarretar uma maior condensação e concentração da umidade relativa do ar. O que vale salientar é que este comportamento pode ter uma correlação maior ou menor dependendo da região e altitude do local em estudo.

Entre as variáveis temperatura mínima e velocidade do vento (x_1 e x_4) não existe correlação significativa, pode se trabalhar diretamente com estas variáveis.

Entre as variáveis temperatura máxima e umidade relativa (x_2 e x_3) não existe correlação, pode se trabalhar diretamente com estas variáveis.

Entre as variáveis temperatura máxima e velocidade do vento (x_2 e x_4) existe uma correlação negativa fraca que não justifica a necessidade de se fazer uma desconexão. É interessante, no entanto, perceber que um aumento na velocidade do vento pode acarretar em uma pequena diminuição na temperatura do local.

Entre as variáveis umidade relativa e velocidade do vento (x_3 e x_4) existe uma correlação negativa fraca desprezando-se a necessidade de se fazer uma desconexão. Observa-se, no entanto, que um aumento na velocidade do vento pode ocasionar uma diminuição da umidade relativa do local.

4.1.5 . Teste de eliminação de variável

Para se descartar uma ou mais variáveis independentes que podem contribuir pouco para a identificação de padrões da rede, se usa o critério de descarte de variáveis redundantes adotado na Análise de Componentes Principais (Mardia et al., 1982) que segue os parâmetros a seguir:

- Determine os autovalores e autovetores da matriz de correlação das variáveis independentes.
- Considere o autovetor correspondente ao menor autovalor (valor absoluto), descarte aí, a variável cujo coeficiente no autovetor for o maior (valor absoluto). É claro que o autovetor com menor autovalor é o menos importante no conjunto.
- O número de variáveis descartadas deve ser igual ao número de autovalores menores ou iguais a 0,70.

Da lista abaixo se identifica qual (is) variável (is) poderá (ão) ser descartada (s):

x_1 = temperatura mínima do ar

x_2 = temperatura máxima do ar

x_3 = umidade relativa do ar

x_4 = velocidade do vento

A partir da matriz de correlação já exposta anteriormente, se gera os autovalores e autovetores:

$$\begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \\ 0,3341 & 0,9615 & 1,0253 & 1,6791 \end{bmatrix}$$

$$\begin{bmatrix} \bar{e}_1 & \bar{e}_2 & \bar{e}_3 & \bar{e}_4 \\ x_1 & 0,7065 & -0,6970 & -0,1223 & -0,0102 \\ x_2 & 0,0813 & -0,0391 & 0,6281 & 0,7729 \\ x_3 & 0,0687 & 0,1937 & -0,7582 & 0,6188 \\ x_4 & -0,6997 & -0,6893 & -0,1250 & 0,1403 \end{bmatrix}$$

Observa-se que o menor valor absoluto em autovalores que seja menor que 0,7 é o λ_1 , e se olhando para a coluna da matriz de autovetores correspondente ao autovalor λ_1 , se identifica o maior valor absoluto correspondendo à variável x_1 .

Portanto, a variável que será eliminada do conjunto será x_1 que corresponde ao valor de temperatura mínima. Isto veio a coincidir com as conclusões anteriormente mencionadas de não haver necessidade na prática de se mensurar os limites superiores e inferiores de temperatura em um determinado intervalo de tempo, basta se medir somente a variável temperatura.

Reduz-se com isto as variáveis para:

x_2 = temperatura do ar (não se chamará mais temperatura máxima do ar, pois será valor de medição contínua)

x_3 = umidade relativa do ar

x_4 = velocidade do vento

4.1.6 . Teste de separação e classificação de duas populações

Este teste pretende verificar se a amostra obtida consegue em termos de classificação uma boa separação entre duas populações distintas que são alvos desta dissertação, a saber: uma classe denominada π_1 onde ocorre geada e uma classe π_2 onde não ocorre geada.

O objetivo é verificar se a amostra possui um erro de classificação baixo o que indica uma boa separação entre as duas populações em estudo, não havendo a necessidade de se efetuar métodos de análise discriminante que possam gerar uma melhor separação dos dados observados.

É empregado o Método de Lachenbruch (Lachenbruch e Mickey, 1968) que é realizado através do seguinte algoritmo:

1. Inicia-se com o grupo π_1 de observações. Retira-se uma observação deste grupo e realiza-se uma função de classificação baseada nas n_1-1 e n_2 observações.
2. Classifica-se a observação suprimida empregando-se a função de classificação construída no passo 1.
3. Repete-se os passos 1 e 2 até que todas as observações do grupo π_1 sejam classificadas. Registra-se $n_1^{(H)}$ como o número de observações classificadas erradas neste grupo.
4. Repete-se os passos 1 a 3 até que todas as observações do grupo π_2 sejam classificadas. Registra-se $n_2^{(H)}$ como o número de observações classificadas erradas neste grupo.

Estima-se as probabilidades de erro de classificação $\hat{P}(2|1)$ e $\hat{P}(1|2)$ de acordo com as formulações:

$$\begin{aligned}\hat{P}(2|1) &= \frac{n_1^{(H)}}{n_1} \\ \hat{P}(1|2) &= \frac{n_2^{(H)}}{n_2}\end{aligned}\tag{4.6}$$

O estimador de erro total de classificação será:

$$\hat{E} = \frac{n_1^{(H)} + n_2^{(H)}}{n_1 + n_2}\tag{4.7}$$

Este método é feito computacionalmente empregando-se como função de classificação:

$$\hat{y} = (\bar{x}_1 - \bar{x}_2)' \cdot S_{ponderado}^{-1} \cdot x\tag{4.8}$$

Os resultados obtidos após se realizar o método são:

Classificação de geada quando não houve = 8,57% (amostra com $n_1 = 70$)

Classificação de clima normal quando houve geada = 3,31% (amostra com $n_2 = 1179$)

Erro Total de classificação = 3,6%

Os dados do Simepar apresentam uma boa separação para classificação e o mesmo tipo de teste é utilizado para se determinar o erro de classificação de cada teste de rede que for aprovado.

4.2 . METODOLOGIA DOS TESTES DE ARQUITETURA DE REDES

Para comprovação do que se estudou na revisão de literatura, se efetuou três tipos de testes com as arquiteturas escolhidas, a saber:

- teste das arquiteturas com 4 variáveis não padronizadas (X_1 , X_2 , X_3 e X_4)
- teste das arquiteturas com 4 variáveis padronizadas (Z_1 , Z_2 , Z_3 e Z_4)
- teste das arquiteturas com 3 variáveis padronizadas (Z_2 , Z_3 e Z_4)

O objetivo destes testes é comprovar a necessidade de um pré-processamento da amostra para que se obtenha uma boa convergência na aprendizagem da rede. Para tanto, se padronizou alguns parâmetros para a realização de todos os testes. Os parâmetros são:

- os valores dos pesos sinápticos e das polarizações são escolhidos de valores aleatórios gerados pelo algoritmo de inicialização da rede neural;
- o modo de treinamento escolhido é por lote;
- o número de iterações é fixado em no máximo de 100 épocas, que foi determinado através de testes preliminares com as arquiteturas escolhidas;
- o tempo máximo para tentativa de convergência é de 12 minutos, que também foi determinado a partir de testes preliminares;
- a função de desempenho empregada é a soma dos erros quadrados e é fixado em 0,02
- a quantidade de testes a serem realizados é de 20 por cada arquitetura exposta a seguir.

As arquiteturas escolhidas são as seguintes:

Foram feitos 6 grupos de testes assim separados:

	Redes c/ 4 entradas não normalizadas
Redes c/ 5 neurônios	Função de ativação “tansig” + Levenberg-Marquardt
	Função de ativação “tansig” + Gradient Descent c/ taxa de aprendizagem adaptativa + $\eta = 0,2$
	Função de ativação “tansig” + Gradient Descent c/ taxa de aprendizagem adaptativa + $\eta = 2$
	Função de ativação “tansig” + Gradient Descent c/ taxa de aprendizagem adaptativa + $\eta = 20$
Redes c/ 10 neurônios	Função de ativação “tansig” + Levenberg-Marquardt
	Função de ativação “tansig” + Gradient Descent c/ taxa de aprendizagem adaptativa + $\eta = 0,2$
	Função de ativação “tansig” + Gradient Descent c/ taxa de aprendizagem adaptativa + $\eta = 2$
	Função de ativação “tansig” + Gradient Descent c/ taxa de aprendizagem adaptativa + $\eta = 20$

	Redes c/ 4 entradas normalizadas
Redes c/ 5 neurônios	Função de ativação “tansig” + Levenberg-Marquardt
	Função de ativação “tansig” + Gradient Descent c/ taxa de aprendizagem adaptativa + $\eta = 0,2$
	Função de ativação “tansig” + Gradient Descent c/ taxa de aprendizagem adaptativa + $\eta = 2$
	Função de ativação “tansig” + Gradient Descent c/ taxa de aprendizagem adaptativa + $\eta = 20$
Redes c/ 10 neurônios	Função de ativação “tansig” + Levenberg-Marquardt
	Função de ativação “tansig” + Gradient Descent c/ taxa de aprendizagem adaptativa + $\eta = 0,2$
	Função de ativação “tansig” + Gradient Descent c/ taxa de aprendizagem adaptativa + $\eta = 2$
	Função de ativação “tansig” + Gradient Descent c/ taxa de aprendizagem adaptativa + $\eta = 20$

Onde:

X_n / Z_n = as variáveis de entrada da rede (X_1/Z_1 = temperatura mínima;

X_2/Z_2 = temperatura máxima; X_3/Z_3 = umidade relativa; X_4/Z_4 = velocidade do ar)

IW_{ij} = pesos sinápticos de entrada (Input Weight)

$1, 2, 3, 4, \dots, n$ = neurônios da camada oculta

W_{kj} = pesos sinápticos de recorrência da saída da camada oculta para a entrada da camada oculta

z^{-1} = atraso unitário

u_n = vetores de espaço de estados

W_{js} = pesos sinápticos entre a camada oculta e o neurônio de saída

S = neurônio de saída

Y_s = resposta de saída do neurônio

Observação: as polarizações b_{j1} (bônus dos neurônios da camada oculta) e b_s (do neurônio de saída) foram suprimidas da representação no diagrama por simplificação.

As fórmulas desta rede se expõem abaixo:

$$Y_s = \varphi_s \left\{ \sum_{j=1}^n W_{js} \cdot Y_j + b_s \right\} \quad p/s = 1 \text{ porque existe apenas um neurônio de saída} \quad (4.9)$$

$n = 5$ neurônios ocultos ou $n = 10$ neurônios ocultos (depende da arquitetura)

A equação (4.9) é a resposta de saída da rede. Agora será detalhado passo a passo, levando-se em conta a camada de entrada, a camada oculta e o banco de atraso unitário, seus fluxos de sinais e as funções de ativação empregadas.

$$Y_j = \varphi_j \left[\left(\sum_{i=1}^4 IW_{ij} \cdot X_i + b_{j1} \right) + \left(\sum_{k=1}^n W_{kj} \cdot u_k \right) \right] \quad j = 1, 2, \dots, n \quad (4.10)$$

$n = 5$ neurônios ocultos ou $n = 10$ neurônios ocultos (depende da arquitetura)

A equação (4.10) demonstra a saída de cada neurônio da camada oculta. O primeiro somatório é a contribuição da camada de entrada e o segundo somatório é a contribuição da recorrência do banco de atraso unitário. Agora se pode formular a saída da rede por combinar (4.10) em (4.11).

$$Y_s = \varphi_s \left\{ \sum_{j=1}^n W_{js} \cdot \varphi_j \left[\left(\sum_{i=1}^4 IW_{ij} \cdot X_i + b_{j1} \right) + \left(\sum_{k=1}^n W_{kj} \cdot u_k \right) \right] + b_s \right\} \quad (4.11)$$

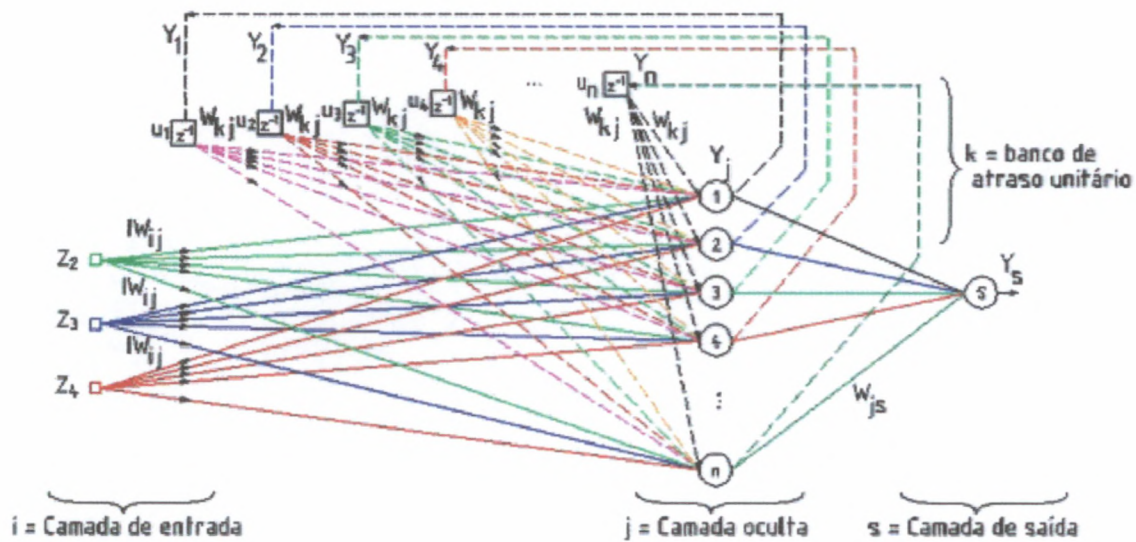
$n = 5$ neurônios ocultos ou $n = 10$ neurônios ocultos (depende da arquitetura)

Sendo $u_k = Y_j(n-1)$, ou seja, o valor de saída de cada neurônio da camada oculta computado anteriormente (atraso unitário). E quando a entrada é padronizada substitui-se X_i por Z_i .

A função de transferência dos neurônios ocultos é o “tansig”, ou seja, $\phi_j(x) = \frac{2}{(1 + e^{-2 \cdot x})} - 1$.

A função de transferência do neurônio de saída é o “logsig”, ou seja, $\phi_s(x) = \frac{1}{(1 + e^{-x})}$.

Figura 4.9 Diagrama da rede de Elman com 3 entradas, n neurônios ocultos e um neurônio de saída



Onde:

Z_n = as variáveis de entrada da rede (Z_2 = temperatura máxima; Z_3 = umidade relativa; Z_4 = velocidade do

IW_{ij} = pesos sinápticos de entrada (Input Weight)

1,2,3,4,..., n = neurônios da camada oculta

W_{kj} = pesos sinápticos de recorrência da saída da camada oculta para a entrada da camada oculta

z^{-1} = atraso unitário

u_n = vetores de espaço de estados

W_{js} = pesos sinápticos entre a camada oculta e o neurônio de saída

S = neurônio de saída

Y_s = resposta de saída do neurônio

Observação: as polarizações b_{j1} (bias dos neurônios da camada oculta) e b_s (do neurônio de saída) foram suprimidos da representação no diagrama por simplificação.

A equação desta rede se expõe abaixo:

$$Y_s = \varphi_s \left\{ \sum_{j=1}^n W_{js} \cdot Y_j + b_s \right\} \quad \text{p/ } s = 1 \text{ porque existe apenas um neurônio de saída} \quad (4.12)$$

$n = 5$ neurônios ocultos ou $n = 10$ neurônios ocultos (depende da arquitetura)

A equação (4.12) é a resposta de saída da rede. Agora será detalhado passo a passo, levando-se em conta a camada de entrada, a camada oculta e o banco de atraso unitário, seus fluxos de sinais e as funções de ativação empregadas.

$$Y_j = \varphi_j \left[\left(\sum_{i=2}^4 IW_{ij} \cdot Z_i + b_{j1} \right) + \left(\sum_{k=1}^n W_{kj} \cdot u_k \right) \right] \quad j = 1, 2, \dots, n \quad (4.13)$$

$n = 5$ neurônios ocultos ou $n = 10$ neurônios ocultos (depende da arquitetura)

A equação (4.13) demonstra a saída de cada neurônio da camada oculta. O primeiro somatório é a contribuição da camada de entrada e o segundo somatório é a contribuição da recorrência do banco de atraso unitário. Agora se pode formular a saída da rede por combinar (4.13) em (4.14).

$$Y_s = \varphi_s \left\{ \sum_{j=1}^n W_{js} \cdot \varphi_j \left[\left(\sum_{i=2}^4 IW_{ij} \cdot X_i + b_{j1} \right) + \left(\sum_{k=1}^n W_{kj} \cdot u_k \right) \right] + b_s \right\} \quad (4.14)$$

$n = 5$ neurônios ocultos ou $n = 10$ neurônios ocultos (depende da arquitetura)

Sendo $u_k = Y_j(n-1)$, ou seja, o valor de saída de cada neurônio da camada oculta computado anteriormente (atraso unitário). E quando a entrada é padronizada substitui-se X_i por Z_i .

A função de transferência dos neurônios ocultos é o “tansig”, ou seja, $\varphi_j(x) = \frac{2}{(1 + e^{-2x})} - 1$.

A função de transferência do neurônio de saída é o “logsig”, ou seja, $\varphi_s(x) = \frac{1}{(1 + e^{-x})}$.

4.2.1 . Teste das Arquiteturas com 4 Variáveis Não Padronizadas (X_1 , X_2 , X_3 e X_4)

Primeiro se efetuou testes das arquiteturas escolhidas com uma amostra composta das 4 variáveis sem se efetuar qualquer tipo de pré-processamento para se concluir os problemas típicos de convergência de uma rede neural.

4.2.1.1 . Rede de Elman com 4 entradas, 5 neurônios ocultos, função de treinamento “Levenberg-Marquardt” (Elman-X-4-5-LM-nn)

Abaixo se registra algumas propriedades comuns para todos os testes executados a partir deste ponto:

Range da primeira entrada: -6,4 a 21

Range da segunda entrada: -6,4 a 29,3

Range da terceira entrada: 48,3571 a 96,7765

Range da quarta entrada: -0,1969 a 4,9044

Legenda:

 Teste paralisado por erro

 Maior tempo de teste

 Menor tempo de teste

(Elman – [X= não normalizado/Z= normalizado] – n° de entradas – n° de neurônios – [algoritmo de aprendizagem= LM/GDA – taxa η] – n° do teste)

Tabela 4.1 – Valores de desempenho dos testes (Elman-X-4-5-LM-01 a 20)

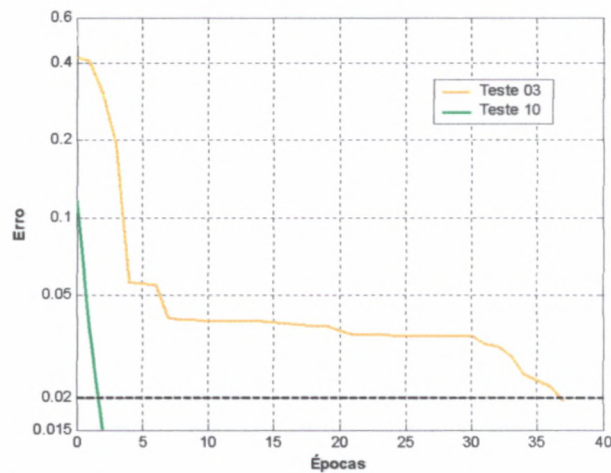
Nº do Teste	Nº de Épocas	Erro	μ	Tempo (min)	Parada
01	4	0,0560	0,0000	1,1932	Gradiente mínimo
02	6	0,0169	0,0100	2,6072	Alvo atingido
03	37	0,0195	0,0000	10,8262	Alvo atingido
04	2	0,0560	0,0001	1,0878	Gradiente mínimo
05	10	0,0163	0,0000	2,9500	Alvo atingido
06	15	0,0187	0,0010	4,7982	Alvo atingido
07	5	0,0198	0,0100	1,6213	Alvo atingido
08	37	0,07943	1e+005	13,0522	Usuário
09	3	0,01948	1000	1,2668	Alvo atingido
10	2	0,015035	0,0100	0,7333	Alvo atingido
11	5	0,014055	0,0100	1,6098	Alvo atingido
12	7	0,017563	0,0001	2,0992	Alvo atingido
13	5	0,013407	0,1000	1,6562	Alvo atingido
14	3	0,016342	10	1,1896	Alvo atingido
15	2	0,018777	10	0,8678	Alvo atingido
16	7	0,018501	1	2,3573	Alvo atingido
17	7	0,017285	0,1000	2,3062	Alvo atingido
18	9	0,0198	0,0010	2,7500	Alvo atingido
19	6	0,0141	0,0100	1,9743	Alvo atingido
20	4	0,0169	0,0100	1,3718	Alvo atingido
Média	8,8	0,0242	50511	2,9159	
Desvio	10,1338	0,0177	22350	3,2407	
Média (sem as paradas)	7,8235	0,0172	60,0742	2,5285	
Desvio (sem as paradas)	8,1795	0,0021	242,2356	2,3444	

Observação:

- 10% dos valores atingiram condição de gradiente mínimo
- 5% dos valores ou uma ocorrência paralisada pelo usuário

O gráfico abaixo apresenta o desempenho de aprendizagem de dois testes expostos acima:

Figura 4.10 Desempenho de erro x épocas de treinamento (Elman-X-4-5-LM-03;10)



4.2.1.2 . Rede de Elman com 4 entradas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptive Learning Rate” (Elman-X-4-5-GDA-0.2-nn)

Tabela 4.2 – Valores de desempenho dos testes (Elman-X-4-5-GDA-0.2-01 a 20)

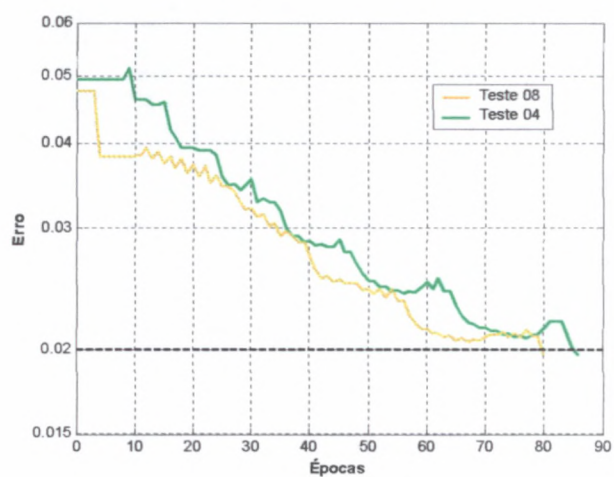
Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	100	0,052904	5,4288 / 6,6806	7,4182	Época
02	100	0,052904	3,4770 / 4,2191	7,6375	Época
03	96	0,052904	3,2666 / 3,2679	8,1240	Gradiente mínimo
04	86	0,01967	0,0112 / 0,0283	6,0137	Alvo atingido
05	100	0,052904	5,4288 / 6,6806	7,4730	Época
06	92	0,052904	3,8772 / 4,3817	7,2825	Gradiente mínimo
07	100	0,024983	1,9974 / 2,0486	10,0303	Época
08	80	0,019608	0,0189 / 0,0284	8,9292	Alvo atingido
09	78	0,052981	1,8821 / 1,5645	12,0950	Usuário
10	98	0,052904	0,9932 / 0,6424	13,0073	Usuário
11	100	0,053014	5,3534 / 6,4828	10,3952	Época
12	95	0,040144	0,6153 / 0,2544	11,6432	Usuário
13	96	0,038279	0,0281 / 0,0319	12,4470	Usuário
14	87	0,036749	0,0976 / 0,0388	11,6750	Usuário
15	81	0,054312	2,6167 / 2,7292	11,9745	Tempo máximo
16	95	0,037585	0,0407 / 0,0235	12,1240	Tempo máximo
17	65	0,055723	1,4565 / 1,2657	12	Tempo máximo
18	87	0,055582	3,2829 / 3,6256	12	Tempo máximo
19	94	0,051952	0,7116 / 0,4718	12	Tempo máximo
20	82	0,052904	2,7168 / 2,8618	12	Tempo máximo
Média	90,6	0,0455	2,2293	10,3135	
Desvio	9,5884	0,0121	3,8703	2,2255	
Média (sem as paradas)	83	0,0196	0,0149	7,4714	
Desvio (sem as paradas)	4,2426	4,3841e-005	0,0285	2,0616	

Observação:

- 25% dos valores atingiram o limite de épocas
- 10% dos valores atingiram gradiente mínimo
- 25% dos valores paralisados pelo usuário
- 30% dos valores atingiram o tempo máximo

O gráfico abaixo apresenta o desempenho de aprendizagem de dois testes expostos acima:

Figura 4.11 Desempenho de erro x épocas de treinamento (Elman-X-4-5-GDA-0.2-04;08)



4.2.1.3 . Rede de Elman com 4 entradas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptive Learning Rate” (Elman-X-4-5-GDA-2-nn)

Tabela 4.3 – Valores de desempenho dos testes (Elman-X-4-5-GDA-2-01 a 20)

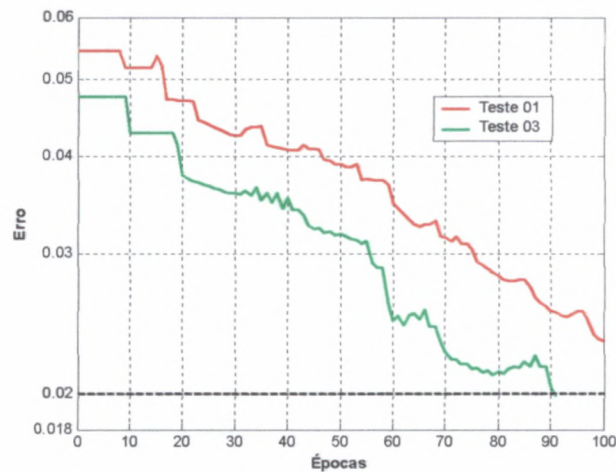
Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	100	0,023336	0,0739 / 0,2704	11,6333	Época
02	54	0,052904	9,9168 / 7,3424	3,7780	Gradiente mínimo
03	91	0,019822	0,0818 / 0,2821	8,0692	goal
04	57	0,052904	10,995 / 8,5342	5,2678	Gradiente mínimo
05	48	0,052904	5,8096 / 2,9846	4,2423	Gradiente mínimo
06	100	0,052904	30,332 / 24,076	7,6767	Época
07	54	0,052904	9,9168 / 7,3424	6,5253	Gradiente mínimo
08	71	0,022137	0,1010 / 0,3167	12	Tempo máximo
09	57	0,052904	10,995 / 8,5342	11,4010	Gradiente mínimo
10	48	0,052904	5,8096 / 2,9846	5,1795	Gradiente mínimo
11	87	0,052904	25,823 / 22,505	12	Tempo máximo
12	88	0,054438	8,5459 / 6,8711	12	Tempo máximo
13	100	0,023117	0,2246 / 0,3314	10,9538	Época
14	81	0,03714	0,1735 / 0,2863	12	Tempo máximo
15	58	0,052904	11,383 / 8,9698	8,8995	Gradiente mínimo
16	65	0,052904	14,565 / 12,657	11,958	Gradiente mínimo
17	92	0,052904	39,766 / 45,889	12	Tempo máximo
18	94	0,03797	0,1010 / 0,2727	12	Tempo máximo
19	88	0,053244	28,955 / 27,482	12,1213	Tempo máximo
20	81	0,053042	23,983 / 22,489	12	Tempo máximo
Média	75,7	0,0453	12,3700	9,5853	
Desvio	18,804	0,0128	21,0531	3,0637	

Observação:

- 15% atingiu o número máximo de épocas
- 40% atingiu o gradiente mínimo
- 40% atingiu o tempo máximo
- os testes que falharam possivelmente ficaram presos em um mínimo local ou uma canaleta e precisariam de um número relativamente maior que 100 épocas para chegar a um resultado favorável.

O gráfico abaixo apresenta o desempenho de aprendizagem de dois testes expostos acima:

Figura 4.12 Desempenho de erro x épocas de treinamento (Elman-X-4-5-GDA-2-01;03)



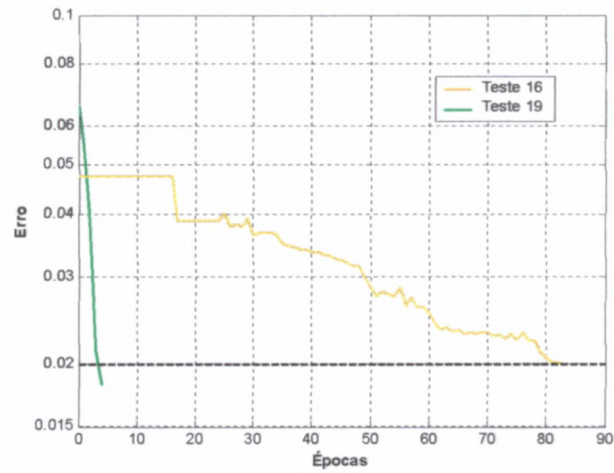
4.2.1.4 . Rede de Elman com 4 entradas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptive Learning Rate” (Elman-X-4-5-GDA-20-nn)

Tabela 4.4 – Valores de desempenho dos testes (Elman-X-4-5-GDA-20-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	8	0,052904	24,5035/ 3,2685	0,8482	Gradiente mínimo
02	11	0,052904	26,5285/ 4,6530	1,2863	Gradiente mínimo
03	97	0,052904	2,1161/ 2,9335	12	Tempo máximo
04	78	0,056039	79,5926/42,776	12	Tempo máximo
05	62	0,056041	130,9/ 109,3	12	Tempo máximo
06	13	0,052904	27,9980/ 5,6918	2,0467	Gradiente mínimo
07	93	0,03102	16,4528/23,205	12	Tempo máximo
08	8	0,052904	24,5035/ 3,2685	1,1877	Gradiente mínimo
09	39	0,056044	60,4/ 33,43	4,5970	Usuário
10	34	0,056015	51,6116/ 25,207	4,5478	Usuário
11	32	0,019926	12,9714/12,08	3,6933	Alvo atingido
12	90	0,055702	104,34/ 55,47	12	Tempo máximo
13	37	0,056023	56,6892/29,909	5	Usuário
14	71	0,056042	180,81/ 169,24	9,2973	Usuário
15	15	0,052904	29,5719/ 6,8339	2,1352	Gradiente mínimo
16	83	0,019908	0,8021/ 2,9662	5,4422	Alvo atingido
17	100	0,053136	77,2964/33,054	7,3305	Época
18	100	0,052884	31,214/ 36,6765	6,8147	Época
19	4	0,018165	22,1025/ 1,7040	0,3383	Alvo atingido
20	100	0,035489	0,7620/ 2,6975	6,9912	Época
Média	53,75	0,0470	53,3767	6,0778	
Desvio	36,9664	0,0136	77,3251	4,2266	

Observação:

- 25% dos testes executados atingiram o gradiente mínimo
- 25% dos testes executados atingiram o tempo máximo
- 20% dos testes paralisaram por ação do usuário
- 15% dos testes atingiram o limite de épocas

Figura 4.13 Desempenho de erro x épocas de treinamento (Elman-X-4-5-GDA-20-16;19)

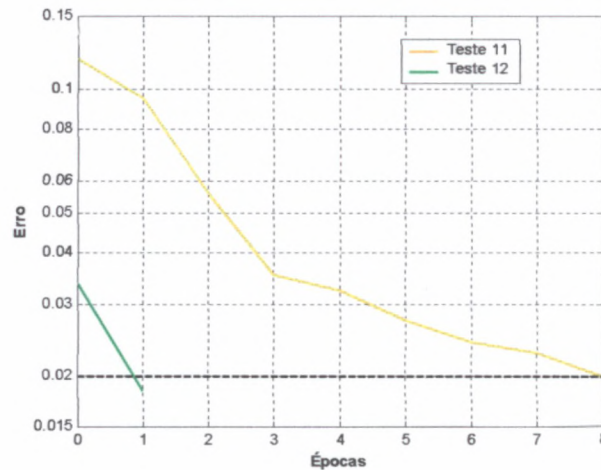
4.2.1.5 . Rede de Elman com 4 entradas, 10 neurônios ocultos, função de treinamento “Levenberg-Marquardt” (Elman-X-4-10-LM-nn)

Tabela 4.5 – Valores de desempenho dos testes (Elman-X-4-10-LM-01 a 20)

Nº do Teste	Nº de Épocas	Erro	μ	Tempo (min)	Parada
01	5	0,0191	100	2,1360	Alvo atingido
02	3	0,0198	1	1,4400	Alvo atingido
03	5	0,0191	0,0100	1,8960	Alvo atingido
04	32	0,056844	0,000001	12,300	Tempo máximo
05	5	0,018618	0,1000	1,9560	Alvo atingido
06	2	0,014312	0,0100	0,9480	Alvo atingido
07	5	0,056045	0,00001	1,7520	Gradiente mínimo
08	3	0,017484	0,0010	1,2240	Alvo atingido
09	3	0,012091	1	1,3680	Alvo atingido
10	7	0,014126	0,0100	2,5320	Alvo atingido
11	8	0,019986	1000	3,1680	Alvo atingido
12	1	0,018451	0,1000	0,7080	Alvo atingido
13	2	0,015989	0,0100	0,9480	Alvo atingido
14	3	0,011454	0,1000	1,3200	Alvo atingido
15	10	0,056045	1e+010	4,0800	μ máximo
16	6	0,01955	0,0100	2,2200	Alvo atingido
17	8	0,018758	100	3,1440	Alvo atingido
18	7	0,019539	10	2,7480	Alvo atingido
19	4	0,017314	0,1000	1,6320	Alvo atingido
20	5	0,012841	0,0100	1,9560	Alvo atingido
Média	6,2	0,022872	5,0005e+008	2,4738	
Desvio	6,5018	0,014658	2,2361e+009	2,4643	
Média (sem as paradas)	4,5294	0,016971	71,321	1,8438	
Desvio (sem as paradas)	2,154	0,0029082	241,56	0,74907	

Observação:

- 5% atingiu o tempo máximo
- 5% atingiu o gradiente mínimo
- 5% atingiu o valor de μ máximo

Figura 4.14 Desempenho de erro x épocas de treinamento (Elman-X-4-10-LM-11;12)

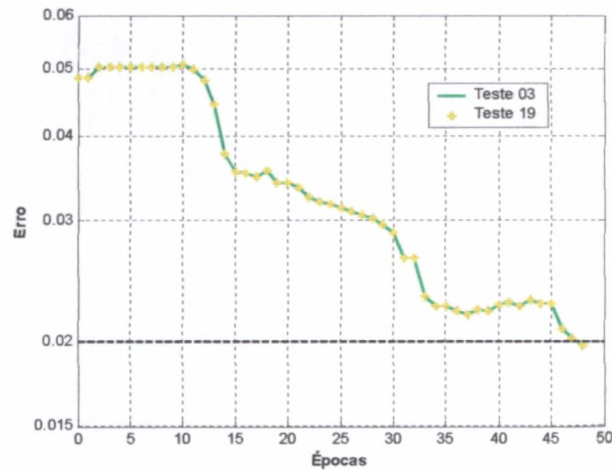
4.2.1.6. Rede de Elman com 4 entradas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptive Learning Rate” (Elman-X-4-10-GDA-0.2-nn)

Tabela 4.6 – Valores de desempenho dos testes (Elman-X-4-10-GDA-0.2-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	100	0,05593	0,056 / 6,6806	7,0388	Época
02	100	0,022679	0,0152 / 0,0329	6,8688	Época
03	48	0,01974	0,0271 / 0,0382	3,2758	Alvo atingido
04	100	0,055982	5,4288 / 6,6806	7,6570	Época
05	100	0,052904	2,2434 / 1,7929	7,6577	Época
06	100	0,05593	5,4288 / 6,6806	7,8653	Época
07	100	0,022679	0,0152 / 0,0329	10,1325	Época
08	48	0,01974	0,0271 / 0,0382	4,7472	Alvo atingido
09	100	0,055982	5,4288 / 6,6806	9,8973	Época
10	100	0,052904	2,2434 / 1,7929	10,0392	Época
11	100	0,045306	0,7043 / 0,8110	9,6265	Época
12	100	0,052903	4,5760 / 5,1718	11,4937	Época
13	100	0,050757	0,6663 / 0,4942	10,2727	Época
14	100	0,043706	0,5636 / 0,1823	11,3090	Época
15	98	0,055979	5,0198 / 6,0827	12,0420	Tempo máximo
16	100	0,055908	5,4288 / 6,6806	11,5193	Época
17	100	0,05593	5,4288 / 6,6806	10,0710	Época
18	100	0,022679	0,0152 / 0,0329	10,3235	Época
19	48	0,01974	0,0271 / 0,0382	5,2917	Alvo atingido
20	97	0,055985	4,8276 / 5,8040	12,1083	Tempo máximo
Média	91,95	0,0437	2,8936	8,9619	
Desvio	18,9583	0,0155	5,0517	2,5417	
Média (sem as paradas)	48	0,0197	0,0271	4,4382	
Desvio (sem as paradas)	0	0	0,0379	1,0429	

Observação: apenas três testes atingiram o valor de erro admissível, contudo, mesmo com escolha aleatória dos pesos das conexões sinápticas e dos biases, observou-se que coincidentemente os valores foram iguais. Isto pode se comprovar ao se ver que tanto o desvio-padrão das épocas quanto o desvio-padrão dos erros foi zero. Também pode se comprovar que sem a normalização dos dados diminui drasticamente a possibilidade de ocorrer convergência no algoritmo, visto que apenas 15% dos testes conseguiram convergência além de se tratarem dos mesmos pontos.

Figura 4.15 Desempenho de erro x épocas de treinamento (Elman-X-4-10-GDA-0.2-03;19)



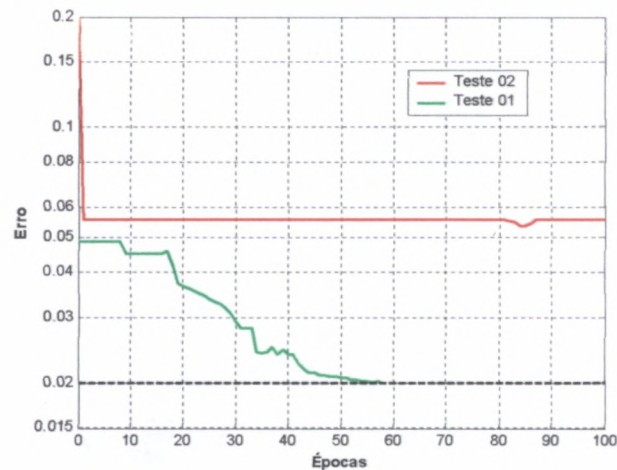
4.2.1.7 . Rede de Elman com 4 entradas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptive Learning Rate” (Elman-X-4-10-GDA-2-nn)

Tabela 4.7 – Valores de desempenho dos testes (Elman-X-4-10-GDA-2-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	58	0,019917	0,1236 / 0,3465	5,5013	Alvo atingido
02	100	0,056002	42,691 / 43,463	7,1728	Época
03	92	0,056045	32,448 / 32,164	6,8363	Gradiente mínimo
04	100	0,055975	46,978 / 52,897	7,5563	Época
05	100	0,052904	6,2275 / 4,9026	7,8870	Época
06	100	0,055975	33,272 / 28,422	7,7692	Época
07	100	0,056038	34,658 / 32,448	7,4470	Época
08	100	0,054306	34,565 / 31,172	7,7732	Época
09	100	0,053389	40,316 / 41,341	7,9055	Época
10	100	0,053295	32,804 / 27,857	8,0123	Época
11	100	0,056044	51,167 / 60,604	8,4313	Época
12	100	0,053294	49,954 / 58,349	8,0710	Época
13	100	0,056043	51,367 / 60,781	8,0045	Época
14	100	0,052904	16,929 / 16,043	8,2845	Época
15	100	0,056044	51,368 / 60,781	8,4688	Época
16	100	0,055834	43,269 / 45,555	8,2930	Época
17	100	0,053144	30,552 / 26,722	7,7140	Época
18	100	0,054911	51,723 / 60,498	8,6797	Época
19	100	0,054659	34,426 / 30,740	9,1027	Época
20	100	0,032969	0,1188 / 0,3369	9,4327	Época
Média	97,5	0,0520	34,9827	7,9172	
Desvio	9,4674	0,0091	43,8505	0,8324	

Observação:

- 90% dos valores atingiram o limite de épocas
- 5% atingiu o gradiente mínimo

Figura 4.16 Desempenho de erro x épocas de treinamento (Elman-X-4-10-GDA-2-01;02)

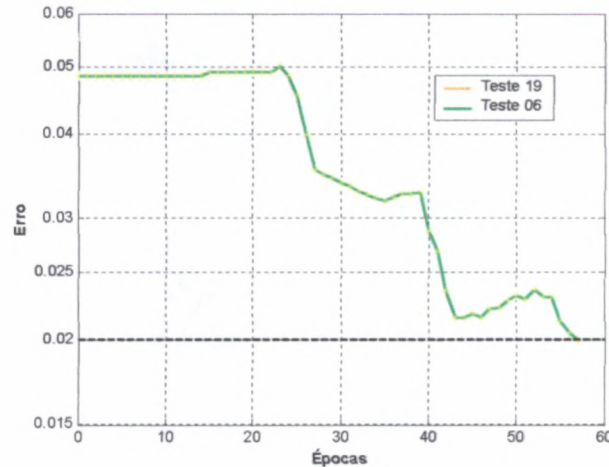
4.2.1.8 . Rede de Elman com 4 entradas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptive Learning Rate” (Elman-X-4-10-GDA-20-nn)

Tabela 4.8 – Valores de desempenho dos testes (Elman-X-4-10-GDA-20-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	41	0,056044	64,3961/ 37,293	3,0868	Usuário
02	2	0,056045	21,0167/ 1,0251	0,2118	Gradiente mínimo
03	94	0,056045	87,573/ 46,8156	7,0380	Gradiente mínimo
04	100	0,0559	164,14/ 122,41	10,3823	Época
05	84	0,024443	0,7912/ 2,9503	12,0620	Tempo máximo
06	57	0,019962	1,1599/ 3,5201	5,41045	Alvo atingido
07	63	0,056045	122,6/ 93,4	8,6383	Gradiente mínimo
08	88	0,05319	315,8/ 331,9	12,0592	Tempo máximo
09	1	0,056045	20,5/ 0,7071	0,2732	Gradiente mínimo
10	100	0,056032	101,4/ 76,21	11,1775	Época
11	1	0,056045	20,5/ 0,7071	0,1627	Gradiente mínimo
12	100	0,053723	135,5/ 115,1	12,0440	Época
13	100	0,055699	74,883/ 34,101	11,2445	Época
14	86	0,056044	316,04/ 345,83	12,1295	Tempo máximo
15	97	0,052904	31,821/ 2,525	12,0680	Tempo máximo
16	74	0,056042	201,77/ 195,5	12,1147	Tempo máximo
17	92	0,056008	155,33/ 123,37	12	Tempo máximo
18	100	0,0237	0,6671/ 2,7192	11,7608	Época
19	57	0,019962	1,1599/ 3,5201	9,0797	Alvo atingido
20	63	0,056045	122,6/ 93,4012	6,9917	Gradiente mínimo
Média	70	0,0488	115,1381	8,4968	
Desvio	34,4154	0,0138	172,5275	4,3980	

Observação:

- 5% dos testes executados paralisados pelo usuário
- 30% dos testes atingiram o gradiente mínimo
- 25% dos testes paralisaram por atingirem o limite máximo de épocas
- 30% dos testes paralisaram ao atingir o tempo limite

Figura 4.17 Desempenho de erro x épocas de treinamento (Elman-X-4-10-GDA-20-06;19)**4.2.2 . Teste das Arquiteturas com 4 Variáveis Padronizadas (Z_1 , Z_2 , Z_3 e Z_4)**

Agora se realizou um pré-processamento para normalizar as 4 variáveis de entrada para se comparar com os testes anteriores e verificar os resultados de convergência de uma rede neural.

4.2.2.1 . Rede de Elman com 4 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Levenberg-Marquardt” (Elman-Z-4-5-LM-nn)

Abaixo se registra algumas propriedades comuns para todos os testes realizados a partir deste ponto:

Range da primeira entrada: -3,9438 a 1,9958

Range da segunda entrada: -3,9362 a 1,7379

Range da terceira entrada: -3,7224 a 2,5678

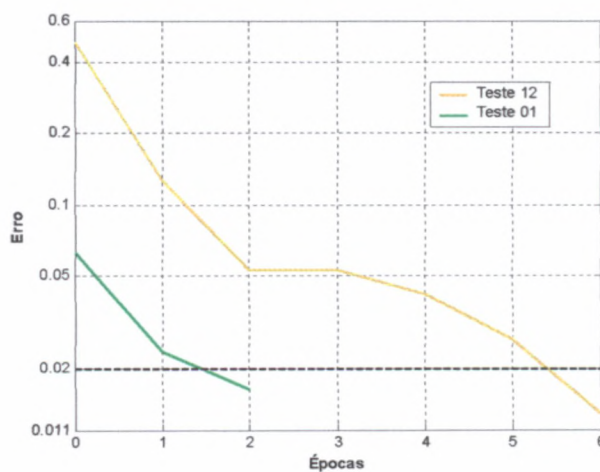
Range da quarta entrada: -4,2837 a 3,3588

Tabela 4.9 – Valores de desempenho dos testes (Elman-Z-4-5-LM-01 a 20)

Nº do Teste	Nº de Épocas	Erro	μ	Tempo (min)	Parada
01	2	0,0164	0,1000	0,9600	Alvo atingido
02	5	0,0064	0,1000	2,8035	Alvo atingido
03	2	0,0168	0,1000	1,3106	Alvo atingido
04	6	0,0156	0,0100	2,0117	Alvo atingido
05	4	0,0139	0,0100	2,0235	Alvo atingido
06	2	0,0126	0,0100	1,5463	Alvo atingido
07	3	0,0178	0,0100	2,0000	Alvo atingido
08	2	0,0163	0,1000	1,5213	Alvo atingido
09	2	0,0160	0,1000	1,4875	Alvo atingido
10	1	0,0182	0,1000	1,1472	Alvo atingido
11	2	0,0164	0,1000	1,7607	Alvo atingido
12	6	0,0128	0,0100	2,9782	Alvo atingido
13	3	0,0188	0,0100	1,9053	Alvo atingido
14	2	0,0199	0,0100	1,3743	Alvo atingido
15	3	0,0156	0,1000	1,8490	Alvo atingido
16	3	0,0162	0,0100	1,7303	Alvo atingido
17	5	0,0196	0,0100	2,0970	Alvo atingido
18	5	0,0175	0,1000	2,1035	Alvo atingido
19	2	0,0153	0,1000	1,1395	Alvo atingido
20	6	0,0169	0,0100	2,4187	Alvo atingido
Média	3,3	0,0159	0,0550	1,8084	
Desvio	1,6255	0,0030	0,0462	0,5301	

Observação: houve 100% de bons resultados neste teste, sendo o teste nº 01 o de menor tempo de aprendizagem e o teste nº 12 o de maior tempo de aprendizagem.

Figura 4.18 Desempenho de erro x épocas de treinamento (Elman-Z-4-5-LM-01;12)



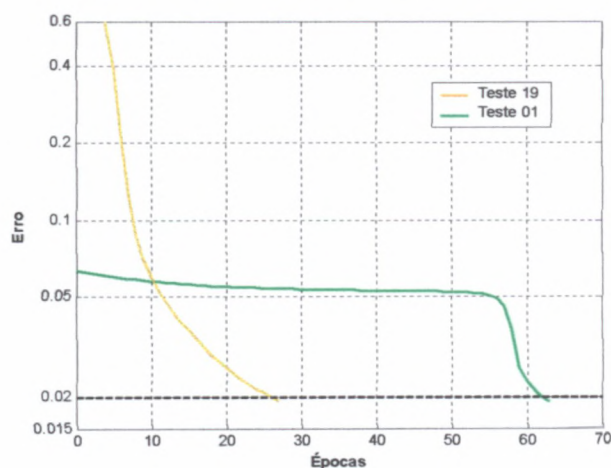
4.2.2.2 . Rede de Elman com 4 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-4-5-GDA-0.2-nn)

Tabela 4.10 – Valores de desempenho dos testes (Elman-Z-4-5-GDA-0.2-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	63	0,019074	1,3565 / 1,1479	5,0687	Alvo atingido
02	54	0,019606	0,9917 / 0,7342	4,2055	Alvo atingido
03	28	0,019563	0,4298 / 0,1757	2,3057	Alvo atingido
04	49	0,019904	0,8374 / 0,5690	4,3667	Alvo atingido
05	61	0,019715	1,2641 / 1,0405	4,9108	Alvo atingido
06	63	0,019074	1,3565 / 1,1479	4,3750	Alvo atingido
07	54	0,019606	0,9917 / 0,7342	3,8053	Alvo atingido
08	49	0,019904	0,8374 / 0,5690	3,4740	Alvo atingido
09	61	0,019715	1,2641 / 1,0405	4,4263	Alvo atingido
10	70	0,018845	1,7435 / 1,6128	4,7972	Alvo atingido
11	29	0,019728	0,4429 / 0,1870	2,0323	Alvo atingido
12	56	0,01995	1,0622 / 0,8118	3,9233	Alvo atingido
13	36	0,019621	0,5493 / 0,2827	2,6085	Alvo atingido
14	41	0,019537	0,6440 / 0,3729	2,9368	Alvo atingido
15	36	0,019801	0,5493 / 0,2827	2,4920	Alvo atingido
16	30	0,01997	0,4565 / 0,1988	2,1168	Alvo atingido
17	31	0,019818	0,4706 / 0,2112	2,2208	Alvo atingido
18	51	0,019648	0,8956 / 0,6306	3,5835	Alvo atingido
19	27	0,01936	0,4172 / 0,1649	1,9125	Alvo atingido
20	31	0,019981	0,4706 / 0,2112	2,1593	Alvo atingido
Média	46	0,01960	0,9612	3,3860	
Desvio	14,1049	3,1627e-004	0,9156	1,0907	

Observação: houve 100% de bons resultados neste teste, sendo o teste nº 19 o de menor tempo de aprendizagem e o teste nº 01 o de maior tempo de aprendizagem,

Figura 4.19 Desempenho de erro x épocas de treinamento (Elman-Z-4-5-GDA-0,2-01;19)



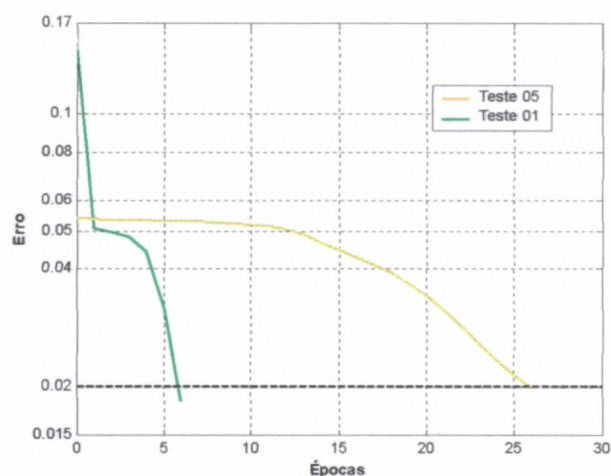
4.2.2.3 . Rede de Elman com 4 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-4-5-GDA-2-nn)

Tabela 4.11 – Valores de desempenho dos testes (Elman-Z-4-5-GDA-2-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	6	0,018383	2,3263 / 0,2449	0,4698	Alvo atingido
02	8	0,019117	2,4503 / 0,3268	0,6250	Alvo atingido
03	14	0,017951	2,8771 / 0,6249	1,0097	Alvo atingido
04	13	0,019474	2,7998 / 0,5692	0,9480	Alvo atingido
05	26	0,019813	4,0496 / 1,5461	1,9042	Alvo atingido
06	6	0,019712	2,3263 / 0,2449	0,4708	Alvo atingido
07	22	0,018175	3,6027 / 1,1798	1,5615	Alvo atingido
08	14	0,01963	2,8771 / 0,6249	1,0095	Alvo atingido
09	14	0,019675	2,8771 / 0,6249	1,0187	Alvo atingido
10	10	0,018942	2,5831 / 0,4169	0,7680	Alvo atingido
11	17	0,018511	3,1258 / 0,8089	1,2147	Alvo atingido
12	13	0,017689	2,7998 / 0,5692	0,9470	Alvo atingido
13	8	0,019112	2,4503 / 0,3268	0,6270	Alvo atingido
14	9	0,019272	2,5156 / 0,3708	0,7532	Alvo atingido
15	11	0,019415	2,6529 / 0,4653	0,8062	Alvo atingido
16	8	0,01911	2,4503 / 0,3268	0,6085	Alvo atingido
17	12	0,019291	2,7251 / 0,5160	0,8917	Alvo atingido
18	23	0,018311	3,7085 / 1,2650	1,6567	Alvo atingido
19	8	0,019141	2,4503 / 0,3268	0,6023	Alvo atingido
20	11	0,019378	2,6529 / 0,4653	0,8045	Alvo atingido
Média	12,65	0,0190	2,9992	0,9348	
Desvio	5,6127	6,2299e-004	0,9743	0,3897	

Observação: houve 100% de bons resultados neste teste, sendo o teste nº 01 o de menor tempo de aprendizagem e o teste nº 05 o de maior tempo de aprendizagem.

Figura 4.20 Desempenho de erro x épocas de treinamento (Elman-Z-4-5-GDA-2-01;05)



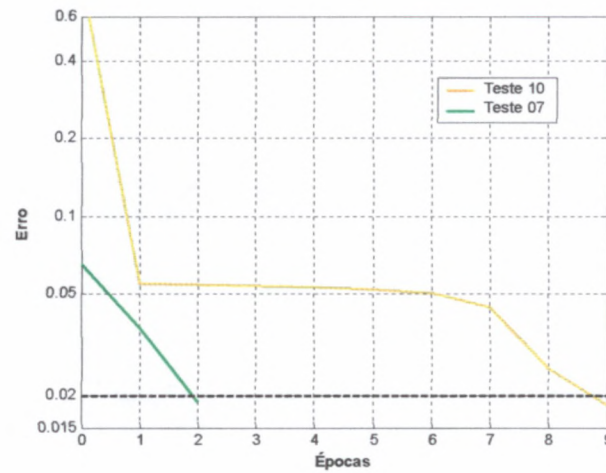
4.2.2.4 . Rede de Elman com 4 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-4-5-GDA-20-nn)

Tabela 4.12 – Valores de desempenho dos testes (Elman-Z-4-5-GDA-20-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	10	0,018922	15,207 / 5,809	1,5240	Alvo atingido
02	14	0,019522	14,457 / 7,628	0,9680	Alvo atingido
03	4	0,017733	22,103 / 1,704	0,3845	Alvo atingido
04	9	0,012632	11,967 / 6,975	0,6097	Alvo atingido
05	89	0,01755	207,16 / 210,05	5,9782	Alvo atingido
06	3	0,017178	21,551 / 1,357	0,2542	Alvo atingido
07	2	0,018846	21,017 / 1,025	0,2460	Alvo atingido
08	40	0,0168	38,788 / 19,61	2,7042	Alvo atingido
09	100	0,046361	313,58 / 343,2	7,2282	Época
10	9	0,018259	25,156 / 3,708	0,6708	Alvo atingido
11	2	0,056045	21,017 / 1,025	0,2782	Gradiente mínimo
12	95	0,017769	236,97 / 253,78	6,8427	Alvo atingido
13	6	0,018445	22,434 / 1,991	0,4708	Alvo atingido
14	1	0,056045	20,5 / 0,707	0,4083	Gradiente mínimo
15	2	0,056045	21,017 / 1,025	0,2948	Gradiente mínimo
16	11	0,018075	17,013 / 6,505	0,7482	Alvo atingido
17	10	0,017397	15,824 / 6,57	0,6928	Alvo atingido
18	87	0,018464	150,4 / 149,5	5,9202	Alvo atingido
19	9	0,01882	13,922 / 7,318	0,7450	Alvo atingido
20	8	0,017805	17,306 / 4,474	0,6055	Alvo atingido
Média	25,55	0,0249	169,3076	1,8787	
Desvio	35,5090	0,0149	237,2756	2,4435	
Média (sem as paradas)	25,375	0,0178	137,5914	1,8353	
Desvio (sem as paradas)	33,3684	0,0015	190,9954	2,2740	

Observação:

- 5% atingiu o máximo de épocas
- 15% atingiu o gradiente mínimo

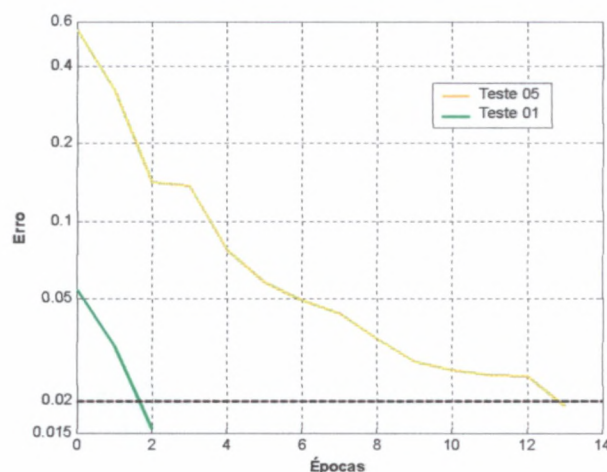
Figura 4.21 Desempenho de erro x épocas de treinamento (Elman-Z-4-5-GDA-20-07;10)

4.2.2.5 . Rede de Elman com 4 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Levenberg-Marquardt” (Elman-Z-4-10-LM-nn)

Tabela 4.13 – Valores de desempenho dos testes (Elman-Z-4-10-LM-01 a 20)

Nº do Teste	Nº de Épocas	Erro	μ	Tempo (min)	Parada
01	2	0,0156	0,1000	1,0273	Alvo atingido
02	8	0,0185	0,0010	2,8168	Alvo atingido
03	4	0,0178	0,0100	1,5858	Alvo atingido
04	3	0,0126	0,0100	1,2802	Alvo atingido
05	13	0,0190	0,0001	4,3668	Alvo atingido
06	7	0,0106	0,1000	2,6202	Alvo atingido
07	8	0,0132	0,0100	3,0055	Alvo atingido
08	6	0,0158	0,0100	2,2760	Alvo atingido
09	3	0,0189	0,0100	1,2710	Alvo atingido
10	3	0,0108	0,1000	1,3282	Alvo atingido
11	10	0,0147	1,0000	3,6280	Alvo atingido
12	4	0,0143	0,0100	1,5875	Alvo atingido
13	4	0,0155	0,0100	1,5792	Alvo atingido
14	4	0,0106	0,0100	1,5802	Alvo atingido
15	3	0,0160	0,0100	1,2482	Alvo atingido
16	5	0,0198	0,0100	1,9148	Alvo atingido
17	3	0,0129	0,1000	1,6503	Alvo atingido
18	7	0,0199	0,0100	2,6702	Alvo atingido
19	6	0,0182	0,0100	2,2575	Alvo atingido
20	4	0,0172	0,1000	1,7012	Alvo atingido
Média	5,35	0,0156	0,0811	2,0697	
Desvio	2,7961	0,0031	0,2200	0,8807	

Observação: houve 100% de bons resultados neste teste, sendo o teste nº 01 o de menor tempo de aprendizagem e o teste nº 05 o de maior tempo de aprendizagem.

Figura 4.22 Desempenho de erro x épocas de treinamento (Elman-Z-4-10-LM-01;05)

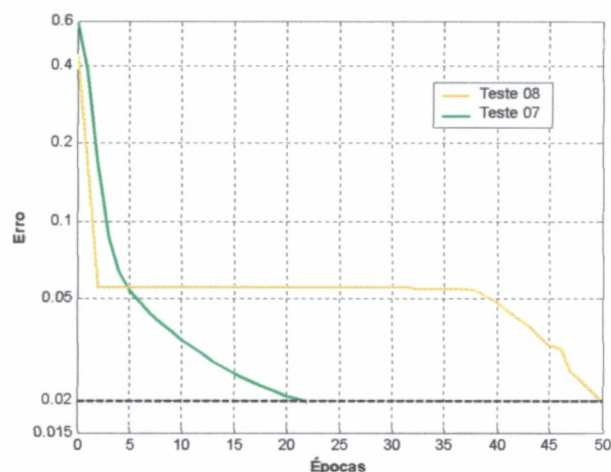
4.2.2.6 . Rede de Elman com 4 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-4-10-GDA-0.2-nn)

Tabela 4.14 – Valores de desempenho dos testes (Elman-Z-4-10-GDA-0.2-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	34	0,019505	0,5161 / 0,2521	3,1990	Alvo atingido
02	33	0,019794	0,5004 / 0,2378	2,4062	Alvo atingido
03	47	0,01976	0,7834 / 0,5130	3,4672	Alvo atingido
04	51	0,018824	0,8956 / 0,6306	3,6460	Alvo atingido
05	22	0,01992	0,3603 / 0,1180	1,6012	Alvo atingido
06	37	0,019274	0,5669 / 0,2991	2,6313	Alvo atingido
07	22	0,019712	0,3603 / 0,1180	1,5865	Alvo atingido
08	50	0,019845	0,8659 / 0,5991	3,7375	Alvo atingido
09	25	0,019509	0,3932 / 0,1448	1,7947	Alvo atingido
10	34	0,019704	0,5161 / 0,2521	2,4073	Alvo atingido
11	27	0,01973	0,4172 / 0,1649	1,9208	Alvo atingido
12	32	0,02	0,4852 / 0,2242	2,3012	Alvo atingido
13	40	0,019808	0,6236 / 0,3532	2,8157	Alvo atingido
14	50	0,019298	0,8659 / 0,5991	3,5490	Alvo atingido
15	44	0,019502	0,6801 / 0,4132	3,2323	Alvo atingido
16	25	0,019918	0,3932 / 0,1448	1,7887	Alvo atingido
17	34	0,019533	0,5161 / 0,2521	2,4073	Alvo atingido
18	26	0,019755	0,4050 / 0,1546	1,8710	Alvo atingido
19	27	0,019252	0,4172 / 0,1649	1,9250	Alvo atingido
20	31	0,019456	0,4050 / 0,1546	2,2062	Alvo atingido
Média	34,55	0,0196	0,5961	2,5247	
Desvio	9,5558	2,8716e-004	0,4162	0,7207	

Observação: houve 100% de bons resultados neste teste, sendo o teste nº 07 o de menor tempo de aprendizagem e o teste nº 08 o de maior tempo de aprendizagem.

Figura 4.23 Desempenho de erro x épocas de treinamento (Elman-Z-4-10-GDA-0.2-07;08)



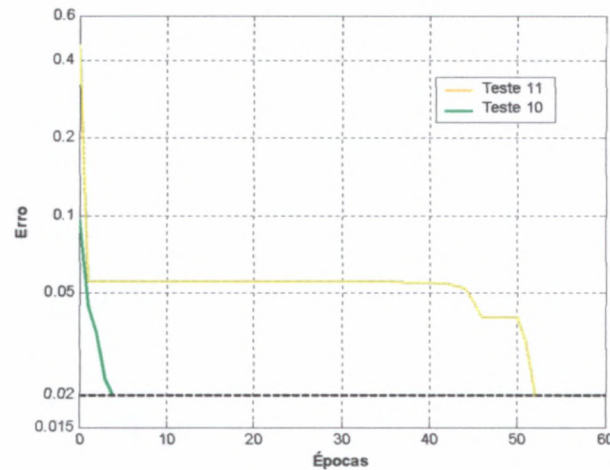
4.2.2.7 . Rede de Elman com 4 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptive Learning Rate” (Elman-Z-4-10-GDA-2-nn)

Tabela 4.15 – Valores de desempenho dos testes (Elman-Z-4-10-GDA-2-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	13	0,017878	2,7998 / 0,5692	0,5118	Alvo atingido
02	12	0,019498	2,7251 / 0,5160	0,8760	Alvo atingido
03	35	0,018523	5,0322 / 2,2288	2,5228	Alvo atingido
04	7	0,019986	2,3873 / 0,2849	0,5773	Alvo atingido
05	6	0,018764	2,3263 / 0,2449	0,5543	Alvo atingido
06	9	0,019627	2,5156 / 0,3708	0,6752	Alvo atingido
07	20	0,017922	3,4018 / 1,0209	1,5887	Alvo atingido
08	30	0,017432	4,5652 / 1,9880	2,1907	Alvo atingido
09	17	0,017399	3,1258 / 0,8089	1,3063	Alvo atingido
10	4	0,019854	2,2103 / 0,1704	0,3823	Alvo atingido
11	52	0,019754	7,5367 / 4,7022	3,6858	Alvo atingido
12	18	0,018911	3,2146 / 0,8763	1,3377	Alvo atingido
13	14	0,019893	2,8771 / 0,6249	1,2522	Alvo atingido
14	6	0,019396	2,3263 / 0,2449	0,5282	Alvo atingido
15	11	0,019844	2,6529 / 0,4653	0,8208	Alvo atingido
16	23	0,017051	3,7085 / 1,2650	1,7557	Alvo atingido
17	12	0,019473	2,6653 / 0,4477	0,8740	Alvo atingido
18	11	0,018259	2,6529 / 0,4653	0,7970	Alvo atingido
19	27	0,019274	4,1716 / 1,6489	1,9303	Alvo atingido
20	17	0,016941	3,1258 / 0,8089	1,2835	Alvo atingido
Média	17,2	0,0188	4,0576	1,2725	
Desvio	11,6646	0,0010	2,6848	0,8281	

Observação: houve 100% de bons resultados neste teste, sendo o teste nº 10 o de menor tempo de aprendizagem e o teste nº 11 o de maior tempo de aprendizagem.

Figura 4.24 Desempenho de erro x épocas de treinamento (Elman-Z-4-10-GDA-2-10;11)



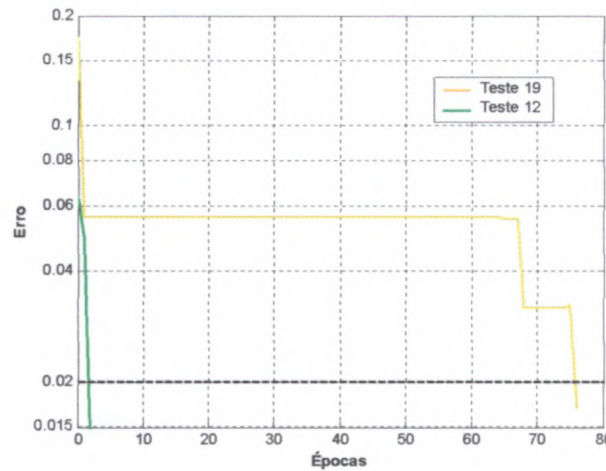
4.2.2.8 . Rede de Elman com 4 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptive Learning Rate” (Elman-Z-4-10-GDA-20-nn)

Tabela 4.16 – Valores de desempenho dos testes (Elman-Z-4-10-GDA-20-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	1	0,056045	20,5 / 0,7071	0,1552	Gradiente mínimo
02	1	0,056045	20,5 / 0,7071	0,3667	Gradiente mínimo
03	11	0,01841	9,1403 / 7,0054	0,7062	Alvo atingido
04	4	0,056045	22,1025 / 1,704	0,3293	Gradiente mínimo
05	1	0,056045	20,5 / 0,7071	0,2668	Gradiente mínimo
06	1	0,056045	20,5 / 0,7071	0,1432	Gradiente mínimo
07	12	0,019027	17,916 / 6,7126	1,4945	Alvo atingido
08	1	0,056045	20,5 / 0,7071	0,1448	Gradiente mínimo
09	2	0,056045	21,017 / 1,0251	0,1845	Gradiente mínimo
10	11	0,016119	10,338 / 5,9306	0,7260	Alvo atingido
11	53	0,015939	65,09 / 41,632	3,5937	Alvo atingido
12	2	0,013579	21,017 / 1,0251	0,1783	Alvo atingido
13	2	0,056045	21,017 / 1,0251	0,2252	Gradiente mínimo
14	1	0,056045	20,5 / 0,7071	0,1480	Gradiente mínimo
15	10	0,017175	20,273 / 4,769	1,3430	Alvo atingido
16	1	0,056045	20,5 / 0,7071	0,1490	Gradiente mínimo
17	1	0,056045	20,5 / 0,7071	0,1368	Gradiente mínimo
18	100	0,056044	542,9 / 668,1	8,3297	Época
19	76	0,016851	142,72 / 123,43	6,2177	Alvo atingido
20	21	0,016183	21,16 / 9,1131	1,6262	Alvo atingido
Média	15,6	0,0403	214,2107	1,3232	
Desvio	27,7970	0,0198	433,6541	2,2267	
Média (sem as paradas)	24,5	0,0167	77,0726	1,9857	
Desvio (sem as paradas)	25,9505	0,0017	95,7397	1,9918	

Observação:

- 55% do total de testes atingiu o gradiente mínimo
- 5% do total de testes paralisou porque atingiu o limite de épocas

Figura 4.25 Desempenho de erro x épocas de treinamento (Elman-Z-4-10-GDA-20-12;19)**4.2.3 . Teste das Arquiteturas com 3 Variáveis Padronizadas (Z_2 , Z_3 e Z_4)**

Em testes preliminares se identificou uma correlação média entre a primeira e a segunda variável. A teoria aconselha como heurística para acelerar a convergência de aprendizagem a descorrelação entre as variáveis fazendo-se uma rotação. Contudo, ao se fazer um teste de descarte de variáveis, se verificou pelo resultado dos autovalores que a variável nº 01 pode ser eliminada sem comprometimento de resposta para a rede.

Nestes testes a seguir vai se comprovar se existe melhoria nas arquiteturas de rede escolhida eliminando-se a primeira variável.

4.2.3.1 . Rede de Elman com 3 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Levenberg-Marquardt” (Elman-Z-3-5-LM-nn)

Abaixo se registra algumas propriedades comuns para todos os testes realizados a partir deste ponto:

Range da primeira entrada: -3,9362 a 1,7379 (Z_2)

Range da segunda entrada: -3,7224 a 2,5678 (Z_3)

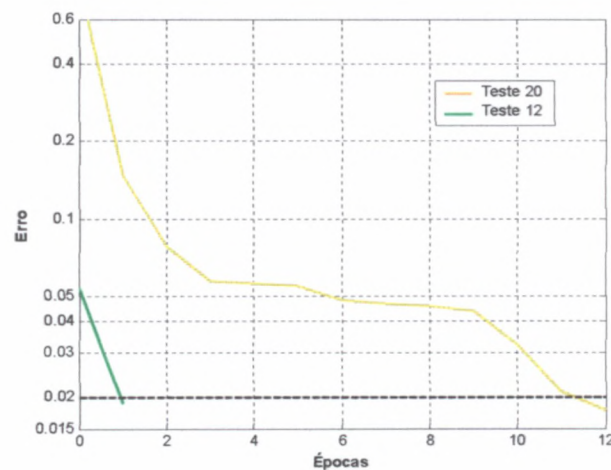
Range da terceira entrada: -4,2837 a 3,3587 (Z_4)

Tabela 4.17 – Valores de desempenho dos testes (Elman-Z-3-5-LM-01 a 20)

Nº do Teste	Nº de Épocas	Erro	μ	Tempo (min)	Parada
01	6	0,0162	0,0100	1,9490	Alvo atingido
02	10	0,0156	0,1000	3,1970	Alvo atingido
03	3	0,0141	0,0100	1,0272	Alvo atingido
04	1	0,0189	0,1000	0,5083	Alvo atingido
05	2	0,0165	0,0100	0,7605	Alvo atingido
06	8	0,0197	0,0010	2,5908	Alvo atingido
07	4	0,0167	0,0100	1,3458	Alvo atingido
08	5	0,0195	0,0100	1,7845	Alvo atingido
09	3	0,0177	0,0100	1,0763	Alvo atingido
10	1	0,0189	0,1000	0,5188	Alvo atingido
11	2	0,0199	0,1000	0,9293	Alvo atingido
12	1	0,0189	0,1000	0,4792	Alvo atingido
13	2	0,0165	0,0100	0,7000	Alvo atingido
14	8	0,0197	0,0010	2,3843	Alvo atingido
15	4	0,0167	0,0100	1,4740	Alvo atingido
16	6	0,0179	0,0100	1,9803	Alvo atingido
17	5	0,0195	0,0100	1,7728	Alvo atingido
18	3	0,0177	0,0100	1,3395	Alvo atingido
19	1	0,0189	0,1000	0,5745	Alvo atingido
20	12	0,0177	0,0010	4,0212	Alvo atingido
Média	4,35	0,0179	0,0357	1,5207	
Desvio	3,1669	0,0016	0,0433	0,9624	

Observação: houve 100% de bons resultados neste teste, sendo o teste nº 12 o de menor tempo de aprendizagem e o teste nº 20 o de maior tempo de aprendizagem.

Figura 4.26 Desempenho de erro x épocas de treinamento (Elman-Z-3-5-LM-12;20)



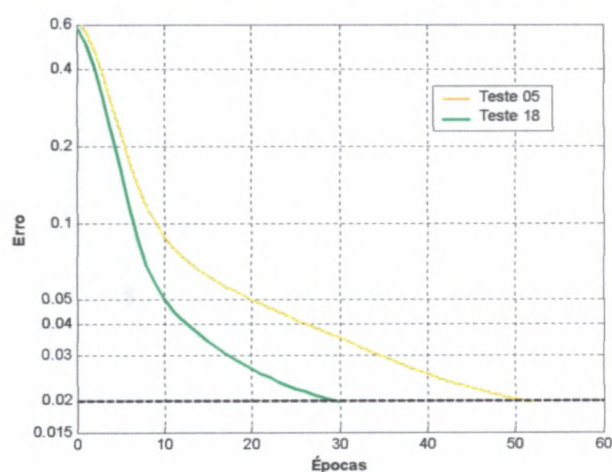
4.2.3.2 . Rede de Elman com 3 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-3-5-GDA-0.2-nn)

Tabela 4.18 – Valores de desempenho dos testes (Elman-Z-3-5-GDA-0.2-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	51	0,019951	0,8956 / 0,6306	5,2387	Alvo atingido
02	40	0,019848	0,6236 / 0,3532	3,5692	Alvo atingido
03	56	0,019898	1,0622 / 0,8118	5,2598	Alvo atingido
04	49	0,019778	0,8374 / 0,5690	4,8470	Alvo atingido
05	52	0,019742	0,9264 / 0,6635	5,8625	Alvo atingido
06	43	0,019837	0,6870 / 0,4153	2,7615	Alvo atingido
07	70	0,019375	1,7435 / 1,6128	4,5835	Alvo atingido
08	47	0,01968	0,7834 / 0,5130	4,6607	Alvo atingido
09	62	0,019956	1,3094 / 1,0929	4,1907	Alvo atingido
10	63	0,019589	1,3039 / 1,0851	4,1282	Alvo atingido
11	36	0,019933	0,5493 / 0,2827	2,2928	Alvo atingido
12	44	0,019647	0,7098 / 0,4381	2,8250	Alvo atingido
13	64	0,019649	1,4055 / 1,2054	4,2960	Alvo atingido
14	31	0,019727	0,4706 / 0,2112	2,0355	Alvo atingido
15	53	0,01988	0,9584 / 0,6981	3,4438	Alvo atingido
16	47	0,019967	0,7834 / 0,5130	3,0553	Alvo atingido
17	50	0,01993	0,8659 / 0,5991	3,2803	Alvo atingido
18	30	0,01967	0,4565 / 0,1988	1,9073	Alvo atingido
19	41	0,019935	0,6440 / 0,3729	2,6982	Alvo atingido
20	42	0,019858	0,6651 / 0,3936	2,7023	Alvo atingido
Média	48,55	0,0198	0,9518	3,6819	
Desvio	10,8165	1,5587e-004	0,8666	1,1597	

Observação: houve 100% de bons resultados neste teste, sendo o teste nº 18 o de menor tempo de aprendizagem e o teste nº 05 o de maior tempo de aprendizagem.

Figura 4.27 Desempenho de erro x épocas de treinamento (Elman-Z-3-5-GDA-0.2-05;18)



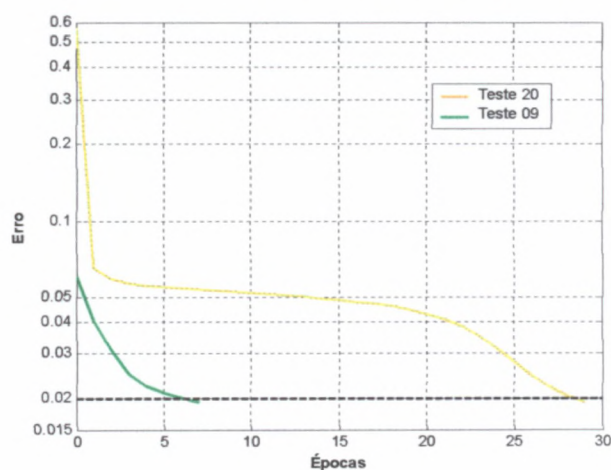
4.2.3.3 . Rede de Elman com 3 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-3-5-GDA-2-nn)

Tabela 4.19 – Valores de desempenho dos testes (Elman-Z-3-5-GDA-2-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	12	0,018914	7,7134 / 4,6907	0,8115	Alvo atingido
02	17	0,018956	6,5385 / 2,1283	1,0785	Alvo atingido
03	15	0,019919	7,7134 / 4,6907	1,0438	Alvo atingido
04	8	0,019004	5,6643 / 2,0731	0,4427	Alvo atingido
05	19	0,019881	8,1388 / 4,7650	1,2053	Alvo atingido
06	14	0,019807	6,2646 / 2,4544	0,8637	Alvo atingido
07	14	0,019745	5,6675 / 2,8666	0,8718	Alvo atingido
08	17	0,019675	8,3849 / 4,7514	1,0532	Alvo atingido
09	7	0,019383	5,1755 / 1,9693	0,4325	Alvo atingido
10	13	0,019963	5,3568 / 2,1146	0,8072	Alvo atingido
11	11	0,018783	4,9976 / 1,1092	0,6833	Alvo atingido
12	12	0,01977	8,1633 / 5,2206	0,7615	Alvo atingido
13	14	0,019558	7,6556 / 5,7480	0,8837	Alvo atingido
14	22	0,019632	6,2369 / 3,9986	1,4282	Alvo atingido
15	18	0,018894	7,7991 / 5,6800	1,1522	Alvo atingido
16	12	0,019381	8,5737 / 5,6609	0,7385	Alvo atingido
17	15	0,01965	6,1572 / 2,8617	0,9472	Alvo atingido
18	9	0,019631	7,7645 / 5,2483	0,5982	Alvo atingido
19	15	0,019942	5,3837 / 3,1541	1,0012	Alvo atingido
20	29	0,019243	5,6876 / 2,2749	1,7773	Alvo atingido
Média	14,65	0,0195	3,1007	0,9291	
Desvio	4,9976	3,9106e-004	1,0205	0,3168	

Observação: houve 100% de bons resultados neste teste, sendo o teste nº 09 o de menor tempo de aprendizagem e o teste nº 20 o de maior tempo de aprendizagem.

Figura 4.28 Desempenho de erro x épocas de treinamento (Elman-Z-3-5-GDA-2-09;20)



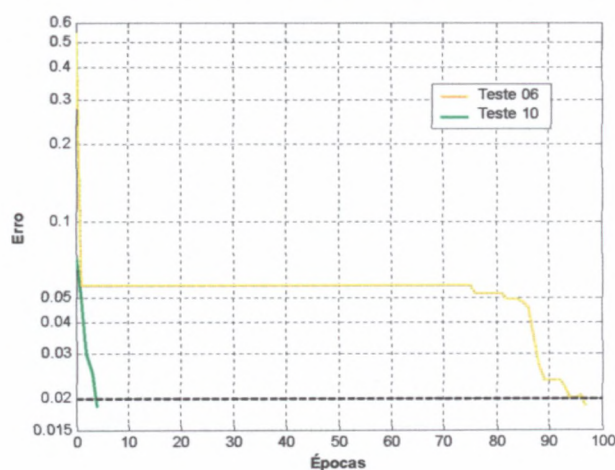
4.2.3.4 . Rede de Elman com 3 entradas padronizadas, 5 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-3-5-GDA-20-nn)

Tabela 4.20 – Valores de desempenho dos testes (Elman-Z-3-5-GDA-20-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	8	0,015523	6,887 / 2,9164	0,4678	Alvo atingido
02	6	0,014718	8,9734 / 3,0326	0,3807	Alvo atingido
03	67	0,018301	64,076 / 70,786	5,4563	Alvo atingido
04	4	0,019268	8,8037 / 4,9297	0,7125	Alvo atingido
05	11	0,017478	10,002 / 6,9552	0,7630	Alvo atingido
06	97	0,018798	168,83 / 177,21	8,4865	Alvo atingido
07	77	0,018984	100,46 / 109,57	7,9073	Alvo atingido
08	6	0,017588	7,7941 / 5,4647	0,4490	Alvo atingido
09	10	0,018079	7,5549 / 4,9911	1,1067	Alvo atingido
10	4	0,018513	12,51 / 4,185	0,1823	Alvo atingido
11	15	0,019865	7,3393 / 5,3800	0,9812	Alvo atingido
12	10	0,017621	7,6326 / 3,4383	0,6313	Alvo atingido
13	11	0,016835	11,76 / 7,454	0,8190	Alvo atingido
14	11	0,019573	7,8948 / 5,8684	1,2553	Alvo atingido
15	91	0,018674	259,33 / 272,62	8,1833	Alvo atingido
16	87	0,018407	163,23 / 178,67	7,6037	Alvo atingido
17	19	0,016404	8,3672 / 7,3486	1,2563	Alvo atingido
18	69	0,017937	78,567 / 81,050	6,1165	Alvo atingido
19	9	0,019303	13,744 / 4,2422	0,6365	Alvo atingido
20	5	0,019845	13,482 / 6,590	0,3710	Alvo atingido
Média	30,85	0,0181	134,2398	2,6883	
Desvio	34,6612	0,0014	173,2490	3,1674	

Observação: houve 100% de bons resultados neste teste, sendo o teste nº 10 o de menor tempo de aprendizagem e o teste nº 06 o de maior tempo de aprendizagem.

Figura 4.29 Desempenho de erro x épocas de treinamento (Elman-Z-3-5-GDA-20-06;10)



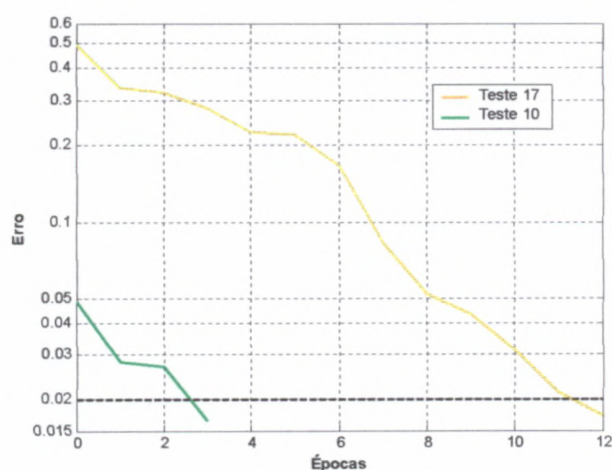
4.2.3.5 . Rede de Elman com 3 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Levenberg-Marquardt” (Elman-Z-3-10-LM-nn)

Tabela 4.21 – Valores de desempenho dos testes (Elman-Z-3-10-LM-01 a 20)

Nº do Teste	Nº de Épocas	Erro	μ	Tempo (min)	Parada
01	4	0,0152	0,1000	1,7542	Alvo atingido
02	9	0,0189	0,0010	3,5103	Alvo atingido
03	6	0,0192	0,0010	1,8552	Alvo atingido
04	5	0,0177	0,0100	1,6898	Alvo atingido
05	9	0,0170	0,0010	2,7208	Alvo atingido
06	12	0,0172	0,0010	3,6137	Alvo atingido
07	3	0,0139	0,1000	1,0635	Alvo atingido
08	5	0,0165	0,0100	1,6637	Alvo atingido
09	4	0,0152	0,1000	1,3983	Alvo atingido
10	3	0,0164	0,1000	1,0615	Alvo atingido
11	8	0,0148	0,0100	2,4647	Alvo atingido
12	3	0,0139	0,1000	1,0660	Alvo atingido
13	5	0,0165	0,0100	1,6875	Alvo atingido
14	4	0,0152	0,1000	1,3802	Alvo atingido
15	3	0,0164	0,1000	1,1657	Alvo atingido
16	10	0,0198	0,0010	3,0252	Alvo atingido
17	12	0,0172	0,0010	3,8115	Alvo atingido
18	3	0,0139	0,1000	1,2313	Alvo atingido
19	5	0,0165	0,0100	1,7177	Alvo atingido
20	4	0,0152	0,1000	1,4280	Alvo atingido
Média	5,85	0,0163	0,0478	1,9654	
Desvio	3,0310	0,0017	0,0486	0,8999	

Observação: houve 100% de bons resultados neste teste, sendo o teste nº 10 o de menor tempo de aprendizagem e o teste nº 17 o de maior tempo de aprendizagem.

Figura 4.30 Desempenho de erro x épocas de treinamento (Elman-Z-3-10-LM-10;17)



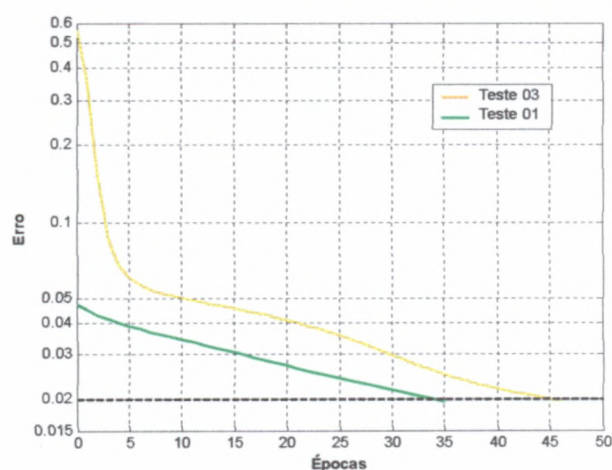
4.2.3.6 . Rede de Elman com 3 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-3-10-GDA-0.2-nn)

Tabela 4.22 – Valores de desempenho dos testes (Elman-Z-3-10-GDA-0.2-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	35	0,019635	0,5324 / 0,2670	2,7315	Alvo atingido
02	37	0,019861	0,5669 / 0,2991	4,3377	Alvo atingido
03	46	0,019673	0,7580 / 0,4868	6,6960	Alvo atingido
04	49	0,019722	0,8374 / 0,5690	6,6167	Alvo atingido
05	47	0,019871	0,7834 / 0,5130	6,5402	Alvo atingido
06	43	0,019968	0,6870 / 0,4153	4,7063	Alvo atingido
07	45	0,019471	0,7334 / 0,4619	5,9160	Alvo atingido
08	43	0,019965	0,6870 / 0,4153	4,8587	Alvo atingido
09	44	0,019978	0,7098 / 0,4381	4,0410	Alvo atingido
10	44	0,019798	0,7098 / 0,4381	4,2083	Alvo atingido
11	37	0,019937	0,5669 / 0,2991	2,9218	Alvo atingido
12	45	0,019631	0,7334 / 0,4619	4,3513	Alvo atingido
13	40	0,019811	0,6236 / 0,3532	4,0367	Alvo atingido
14	50	0,019756	0,8659 / 0,5991	5,3837	Alvo atingido
15	33	0,019551	0,5004 / 0,2378	4,1083	Alvo atingido
16	38	0,019836	0,5851 / 0,3163	3,8333	Alvo atingido
17	32	0,019745	0,4852 / 0,2242	3,9035	Alvo atingido
18	33	0,01992	0,5004 / 0,2378	2,8062	Alvo atingido
19	30	0,019866	0,4565 / 0,1988	2,7698	Alvo atingido
20	36	0,019595	0,5493 / 0,2827	3,0273	Alvo atingido
Média	40,35	0,0198	0,6605	4,3897	
Desvio	6,0199	1,4897e-004	0,42187	1,2843	

Observação: houve 100% de bons resultados neste teste, sendo o teste nº 01 o de menor tempo de aprendizagem e o teste nº 03 o de maior tempo de aprendizagem.

Figura 4.31 Desempenho de erro x épocas de treinamento (Elman-Z-3-10-GDA-0.2-01;03)



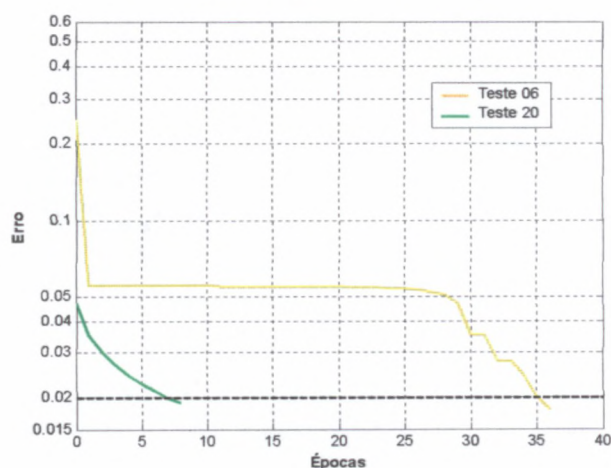
4.2.3.7 . Rede de Elman com 3 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptative Learning Rate” (Elman-Z-3-10-GDA-2-nn)

Tabela 4.23 – Valores de desempenho dos testes (Elman-Z-3-10-GDA-2-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	14	0,01932	2,8771 / 0,6249	1,2323	Alvo atingido
02	11	0,019122	2,6529 / 0,4653	0,7668	Alvo atingido
03	15	0,019845	2,9572 / 0,6834	1,0442	Alvo atingido
04	12	0,019054	2,7251 / 0,5160	0,7995	Alvo atingido
05	49	0,019695	5,3925 / 2,3483	3,5605	Alvo atingido
06	36	0,017935	4,6782 / 1,8557	3,8323	Alvo atingido
07	15	0,01925	2,9572 / 0,6834	1,4462	Alvo atingido
08	14	0,018593	2,8771 / 0,6249	0,9825	Alvo atingido
09	21	0,019037	3,5005 / 1,0985	1,8490	Alvo atingido
10	12	0,019469	2,7251 / 0,5160	0,8208	Alvo atingido
11	10	0,019869	2,5831 / 0,4169	0,7375	Alvo atingido
12	13	0,01996	2,7998 / 0,5692	0,8938	Alvo atingido
13	21	0,019057	3,5005 / 1,0985	1,4740	Alvo atingido
14	12	0,019626	2,7251 / 0,5160	0,8353	Alvo atingido
15	11	0,019829	2,1053 / 0,2432	0,7460	Alvo atingido
16	18	0,019886	3,2146 / 0,8763	1,2908	Alvo atingido
17	22	0,019776	3,6027 / 1,1798	1,5500	Alvo atingido
18	20	0,019265	3,4018 / 1,0209	1,6292	Alvo atingido
19	15	0,019267	2,9572 / 0,6834	1,0500	Alvo atingido
20	8	0,019036	2,4503 / 0,3268	0,4573	Alvo atingido
Média	17,45	0,0193	3,5052	1,3499	
Desvio	9,6544	5,0345e-004	1,5648	0,8799	

Observação: houve 100% de bons resultados neste teste, sendo o teste nº 20 o de menor tempo de aprendizagem e o teste nº 06 o de maior tempo de aprendizagem.

Figura 4.32 Desempenho de erro x épocas de treinamento (Elman-Z-3-10-GDA-2-06;20)



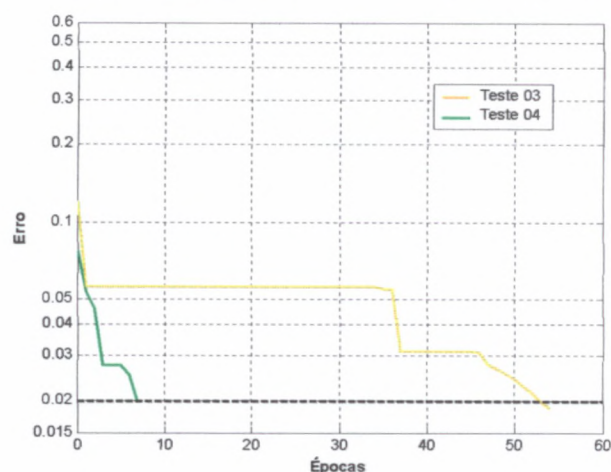
4.2.3.8 . Rede de Elman com 3 entradas padronizadas, 10 neurônios ocultos, função de treinamento “Gradient Descent Adaptive Learning Rate” (Elman-Z-3-10-GDA-20-nn)

Tabela 4.24 – Valores de desempenho dos testes (Elman-Z-3-10-GDA-20-01 a 20)

Nº do Teste	Nº de Épocas	Erro	LR média/desvio	Tempo (min)	Parada
01	9	0,018666	16,51 / 4,5934	0,5032	Alvo atingido
02	7	0,01962	14,1415 / 6,235	0,4033	Alvo atingido
03	54	0,018803	41,9536 / 30,24	4,1147	Alvo atingido
04	7	0,019721	17,2717 / 4,88	0,3718	Alvo atingido
05	10	0,016815	12,049 / 6,4886	0,5303	Alvo atingido
06	20	0,019244	9,1476 / 7,7272	1,2962	Alvo atingido
07	7	0,019609	21,214 / 2,7764	0,3948	Alvo atingido
08	10	0,016392	17,638 / 6,4331	0,5617	Alvo atingido
09	9	0,017643	22,199 / 2,6724	0,6012	Alvo atingido
10	1	0,056045	20,5 / 0,7071	0,1022	Gradiente mínimo
11	42	0,017676	49,481 / 24,166	2,7520	Alvo atingido
12	8	0,017816	13,543 / 6,1022	0,4095	Alvo atingido
13	100	0,056044	542,9 / 668,1	7,2198	Época
14	1	0,056045	20,5 / 0,7071	0,1595	Gradiente mínimo
15	1	0,056045	20,5 / 0,7071	0,0823	Gradiente mínimo
16	19	0,018731	21,73 / 8,1319	1,1667	Alvo atingido
17	19	0,019451	28,32 / 6,7733	1,1865	Alvo atingido
18	30	0,056045	32,673 / 8,8159	1,9283	Gradiente mínimo
19	1	0,056045	20,5 / 0,7071	0,2760	Gradiente mínimo
20	1	0,056045	20,5 / 0,7071	0,0605	Gradiente mínimo
Média	17,8	0,0316	167,3754	1,2060	
Desvio	23,9991	0,0184	413,8691	1,7421	
Média (sem as paradas)	17	0,0185	29,4221	1,0994	
Desvio (sem as paradas)	14,7705	0,0011	23,4171	1,1209	

Observação:

- 30% atingiu o gradiente mínimo
- 5% atingiu o limite de épocas

Figura 4.33 Desempenho de erro x épocas de treinamento (Elman-Z-3-10-GDA-20-03;04)

4.2.4 . Comparativo dos Melhores Resultados dos Testes

Faz-se uma coleta dos melhores resultados obtidos entre os 480 testes realizados. Efetua-se uma tabela comparativa aonde se coloca os valores em função dos menores tempos de execução do algoritmo de aprendizagem. A partir desta tabela, seleciona-se o melhor desempenho para eleger a estrutura de rede desejada. Todos os testes foram executados em um Notebook Acer TravelMate 506DX, processador Celeron 433 MHz, 128 Mb de RAM.

Tabela 4.25 – Tabela comparativa dos melhores resultados

Nº	Código do Teste	Nº de épocas	Erro	Tempo (min)
01	Elman-Z-3-5-GDA-20-10	4	0,018513	0,1823
02	Elman-Z-4-10-GDA-2-10	4	0,019854	0,3823
03	Elman-Z-3-5-GDA-2-09	7	0,019383	0,4325
04	Elman-Z-3-10-GDA-2-20	8	0,019036	0,4573
05	Elman-Z-4-5-GDA-2-19	6	0,018383	0,4698
06	Elman-Z-3-5-LM-12	1	0,0189	0,4792
07	Elman-Z-4-5-LM-01	2	0,0164	0,96
08	Elman-Z-4-10-LM-01	2	0,0156	1,0273
09	Elman-Z-3-10-LM-10	3	0,0164	1,0615
10	Elman-Z-4-10-GDA-0.2-07	22	0,019712	1,5865
11	Elman-Z-3-5-GDA-0.2-18	30	0,01967	1,9073
12	Elman-Z-4-5-GDA-0.2-19	27	0,01936	1,9125
13	Elman-Z-3-10-GDA-0.2-01	35	0,019635	2,7315

Dos 480 testes realizados existem 13 testes que apresentaram os melhores resultados em termos de erro e tempo de processamento. Para se efetuar a escolha da melhor arquitetura para a aplicação em estudo, primeiro se efetua um teste com o Método de Lachenbruch (Lachenbruch e Mickey,

1968) que já foi explanado no tópico 4.1.6 após se aplicar o pós-processamento conforme é descrito no próximo tópico.

4.3 . PÓS-PROCESSAMENTO DA SAÍDA DA REDE

Precisa-se fazer um pós-processamento do sinal de saída da rede para que se tenha uma resposta de comportamento dicotômico, ou seja, 0 para falta de geada e 1 para ocorrência de geada.

Como a rede criada utiliza como função de transferência o algoritmo “logsig”, que pode excursionar o seu valor entre 0 e 1, faz-se necessária a criação de um limiar (“*threshold*”) que gere a resposta dicotômica.

Estipulou-se para o limiar o valor de 0,5 que fica no ponto médio da função de transferência “logsig”. Desta forma, o sinal de saída do neurônio será processado de acordo com a formulação a seguir:

$$\begin{aligned} Y &= 1, \quad \forall Y_s \geq 0,5 \\ Y &= 0, \quad \forall Y_s < 0,5 \end{aligned} \quad (4.15)$$

A tabela abaixo resume os valores de erro parcial e total de classificação das treze arquiteturas que apresentaram melhor desempenho durante o treinamento.

Tabela 4.26 – Tabela de erros de classificação

Nº	Código do Teste	Classificar Geada quando não houve	Classificar Normal quando houve geada	Erro Total de Classificação
01	Elman-Z-3-5-GDA-20-10	0%	1,72%	1,61%
02	Elman-Z-3-5-LM-12	0%	2,05%	1,92%
03	Elman-Z-4-5-GDA-2-19	0%	2,21%	2,08%
04	Elman-Z-4-5-GDA-0.2-19	1,37%	2,38%	2,32%
05	Elman-Z-4-10-GDA-2-10	0%	2,87%	2,73%
06	Elman-Z-3-5-GDA-2-09	0%	2,96%	2,80%
07	Elman-Z-4-10-GDA-0.2-07	2,99%	2,96%	2,96%
08	Elman-Z-3-5-GDA-0.2-18	0%	3,37%	2,96%
09	Elman-Z-4-10-LM-01	5,97%	3,13%	3,28%
10	Elman-Z-4-5-LM-01	0%	3,61%	3,44%
11	Elman-Z-3-10-GDA-2-20	1,64%	3,62%	3,52%
12	Elman-Z-3-10-GDA-0.2-01	3,28%	3,70%	3,68%
13	Elman-Z-3-10-LM-10	0%	4,18%	4%

Efetuada-se a escolha da arquitetura em função dos erros de classificação a configuração que apresentou o melhor resultado foi o teste número 01 que é composto de uma rede de Elman com

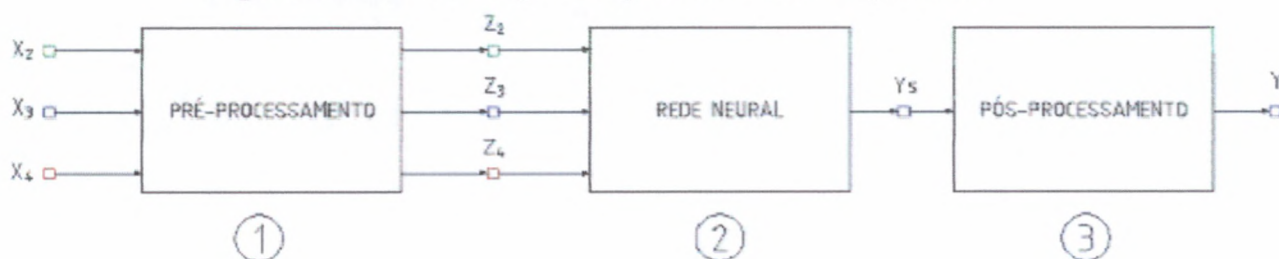
três entradas de variáveis normalizadas, 5 neurônios na camada oculta, 5 realimentações com atraso unitário e que foi treinado com algoritmo “*Gradient Descent with Adaptive Learning Rate*” com taxa de aprendizagem inicial igual a 20. Os valores finais de pesos e polarizações está na tabela I no apêndice I.

4.4 . ARQUITETURA FINAL COM OS ESTÁGIOS DE PRÉ E PÓS-PROCESSAMENTO

Agora se expõe um diagrama de blocos representando o percurso dos sinais de entrada, de acordo com a seguinte seqüência:

1. passando pelo pré-processamento aonde ocorre a padronização dos sinais conforme a Equação (4.2);
2. os sinais padronizados são processados pela rede de Elman conforme a Equação (4.14);
3. a saída da rede neural passa por um pós-processamento para gerar saída dicotômica conforme a Equação (4.15).

Figura 4.34 Módulos que compõe o autômato de previsão



5. CONCLUSÃO

Os testes comprovam a teoria de convergência sobre a necessidade de padronização das variáveis de entrada (LeCun, 1993), ou seja, um pré-processamento das variáveis de entrada em que se garante uma média igual a zero e um desvio-padrão unitário. Os testes sem padronização se mostraram ineficientes em atingir o alvo estipulado.

Uma melhoria a ser feita é uma maior coleta de dados das variáveis pertinentes, a saber: temperatura, umidade relativa do ar e velocidade do vento. Os dados fornecidos pelo Simepar são diários e o comportamento climático exigiria uma coleta de dados em intervalos que poderiam ser de 15 minutos, de preferência empregando-se estações posicionadas em pontos de maior incidência do fenômeno como vales e/ou depressões entre colinas e regiões montanhosas, onde ocorre menor insolação ao longo do dia e, portanto, maior perda de calor do solo para o ar em sua volta ao chegar os períodos mais frios da madrugada.

A partir desta coleta de dados com varredura de 15 minutos se justificaria alguns testes complementares:

- dividir os dados em três blocos para treinamento, teste e validação;
- realizar testes de treinamento com os dados divididos para todas as treze arquiteturas que apresentaram os melhores desempenhos;
- efetuar novos cálculos de erro de classificação empregando o Método de Lachenbruch.

A variável temperatura apresenta na prática um comportamento de variação lenta ao longo de um período de 24 horas, justificando se realizar novos testes de arquitetura variando-se o valor do atraso de realimentação para a camada de entrada. Por exemplo, podia se realimentar a saída dos neurônios ocultos a cada quatro leituras das variáveis de entrada e fazer os teste de treinamento, depois mudar a configuração para oito varreduras, testar e assim por diante. Depois, realizar testes de erro de classificação e comparar com as configurações realizadas neste trabalho para avaliar se houve melhorias de reconhecimento dos padrões.

De acordo com o teste realizado no tópico 4.1.6, os erros de classificação dos dados fornecidos são de geada quando não houve igual a 8,57% , clima normal quando houve geada igual a 3,31% e erro total de classificação igual a 3,6%.

Para diminuir os erros de classificação teria que se empregar métodos como a análise de componentes principais, análise discriminante, etc para realizar uma melhor separação entre os dois grupos de classificação, a saber presença ou ausência de formação de geada.

A proposta de se realizar a classificação destes dois grupos empregando-se uma rede neural apresentou na melhor situação, de acordo com a tabela 4.26, erro de classificação de geada quando não ocorreu nula, erro de classificação de clima normal quando houve geada igual a 1,72% e erro total de classificação igual a 1,61%.

Para a pior situação entre as treze melhores arquiteturas, o erro de classificação de geada quando não ocorreu é nulo, o erro de classificação de clima normal quando houve geada igual a 4,18% e o erro total de classificação igual a 4%.

Com estes valores pode se interpretar que a rede neural conseguiu um bom desempenho em classificar os dois grupos em relação ao erro de classificação obtido nos dados originais, possibilitando uma boa separação entre os dois padrões a serem observados. Conclui-se que esta arquitetura de rede pode ser empregada para a construção de um autômato capaz de prever a formação de geada e se utilizar desta informação de saída dicotômica para disparar um sistema de sprinkler.

Seria interessante a montagem de um protótipo baseado em microcontrolador que possa conter em programação interna o algoritmo que realiza a função da arquitetura do teste **Elman-Z-3-5-GDA-20-10** com os valores de pesos e polarizações definidos na Tabela I do apêndice.

Uma vez construído o protótipo, testes adicionais precisam ser realizados para definir o momento ideal de disparo dos sprinklers e o período de manutenção do esguicho. Estes testes dependem de implementação prática por causa da vazão do esguicho, número de pontos de sprinkler e área de cobertura nos cultivos a serem protegidos.

Aqui fica a sugestão de se repassar este trabalho para órgãos competentes, tais como Simepar, Embrapa e CEFET-PR (setor de agronomia), sob orientação dos professores da UFPR, para se criar o autômato para testes em campo na região de Palmas, P.R.

6 . REFERÊNCIA BIBLIOGRÁFICA:

Antikienen, M., Griffith, M., Hang, J., Hong, W.-C., Yang, D.S.C., Pihakaski-Maunsbach, K., 1996. **Immunolocalization of antifreeze proteins in winter rye leaves, crowns, and roots by tissue printing**. Plant Physiol. 110, 845±857.

Ashworth e Davis, 1986, **Ice formation in woody plants under field conditions**. Hort Science 21: 1233±1234.

B . Krasovitski ; E . Kimmel ; I . Amir, 1995, **simulation of foam protection of plants against frost** , 1995 , 61 : 155 – 163

Bishop, 1995, **Neural Networks for Pattern Recognition**. Oxford: University Press.

Elman, J. L., 1990, **Finding structure in time**, Cognitive Science, vol. 14, pp. 179-211.

Fahlman, 1988, **Faster-learning variations on back-propagation: an empirical study**. In D. Touretzky, G.E. Hinton and T.J. Sejnowski (Eds.), Proceedings of the 1988 Connectionist Models Summer School, 38-51. San Mateo, CA: Morgan Kaufmann.

Fausett,1994, **Fundamentals of Neural Networks**. New York: Prentice Hall.

Franks, 1985, **Biophysics and biochemistry at low temperatures**. Cambridge: Cambridge University Press.

Giese, M.A. 2000. **Neural model for the recognition of biological motion**. From G. Barattoff and H. Neumann (Eds) *Dynamische Perzeption* Infix Verlag, Berlin, pp. 105-110.

Gil, Paulo, Henriques, J., Dourado, A. and Duarte-Ramos, H., 1999. **Non-linear predictive control based on a recurrent neural network**. Presented at ESIT'99, Orthodox Academy of Crete, June 3-4, 1999.

Haykin, 2001, **Redes Neurais: Princípios e prática**. Trad. Paulo Martins Engel – 2.ed. – Porto Alegre: Bookman.

Hornby, G. S., Lipson, H., & Pollack, J.B., 2001. **Evolution of generative design systems for modular physical robots**. International Conference on Robotics and Automation.

Jacobs, 1988, **Increased Rates of Convergence Through Learning Rate Adaptation**. *Neural Networks* 1 (4), 295-307.

Lachenbruch, P.A., and M. R. Mickey, 1968. **Estimation of Error Rates in Discriminant Analysis**. *Technometrics*, 10, n° 1, 1-11.

LeCun, Y., 1993, **Efficient Learning and Second-order Methods**, A Tutorial at NIPS 93, Denver.

Levenberg, 1944, **A method for the solution of certain non-linear problems in least squares**. *Quarterly Journal of Applied Mathematics* II (2), 164-168.

Mardia, K.V., J.T. Kent and J.M. Bibby, 1982, **Multivariate Analysis**. Academic Press: London.

Marquardt, 1963, **An algorithm for least-squares estimation of non-linear parameters**. *Journal of the Society of Industrial and Applied Mathematics* 11 (2), 431-441.

Patterson, 1996, **Artificial Neural Networks**. Singapore: Prentice Hall.

Pearce e Fuller, 2001, **of barley (Hordeum) studied by infrared video thermography**. *Plant Physiology* 125: 227±240.

Rumelhart et. al., 1986, **Learning internal representations by error propagation**. In D.E. Rumelhart, J.L. McClelland (Eds.), **Parallel Distributed Processing, Vol 1**. Cambridge, MA: MIT Press.

Rumelhart e McClelland, 1986, **Parallel Distributed Processing, Vol 1**. Cambridge, MA: MIT Press.

Shepherd, 1997, **Second-Order Methods for Neural Networks**. New York: Springer.

APÊNDICE I:

A tabela a seguir é obtida no treinamento da rede Elman com a arquitetura denominada Elman-Z-3-5-GDA-20-10.

Tabela I – Valores dos pesos e polarizações finais do teste 10 (Elman-Z-3-5-GDA-20-10)

Input Weights (IW_y)	Camada de entrada				
	Entrada 1 (i)	Entrada 2	Entrada 3		
Neurônio oculto 1 (j)	0,161317	-0,28809	0,093491		
Neurônio oculto 2	-0,3322	-0,20318	0,180007		
Neurônio oculto 3	-1,1155	0,02308	-0,14096		
Neurônio oculto 4	-1,10358	-0,37014	0,170593		
Neurônio oculto 5	1,177294	-0,01779	0,011554		
Weight (W_{kj})	Banco de atraso unitários p/ camada de entrada				
	$Z^{-1}(k=1)$	$Z^{-1}(2)$	$Z^{-1}(3)$	$Z^{-1}(4)$	$Z^{-1}(5)$
Neurônio oculto 1 (j)	0,688751	-0,67277	-0,89082	-0,2902	-0,29837
Neurônio oculto 2	1,406393	0,857133	-0,58451	0,062479	-0,58538
Neurônio oculto 3	0,85802	-0,24101	-0,58592	-0,16547	-1,15001
Neurônio oculto 4	1,073124	-0,06377	0,830352	-0,17557	0,076616
Neurônio oculto 5	-0,50977	-0,13775	0,688325	0,672008	-0,76858
Weight (W_{js})	Camada de neurônios ocultos para neurônio de saída				
	Neur. 1 (j)	Neur. 2	Neur. 3	Neur. 4	Neur. 5
Neurônio de saída (s)	-0,16601	1,4392	1,425534	1,797436	-1,54118
Polarização (b_{j1})	Camada de neurônios ocultos				
	Neur. 1 (j)	Neur. 2	Neur. 3	Neur. 4	Neur. 5
	-1,63817	-1,10478	-0,30914	-1,20804	1,812951
Polarização (b_s)	Neurônio de saída				
	-0,3627				

Tabela II – Valores de desempenho do teste 10 (Elman-Z-3-5-GDA-20-10)

Nº de Épocas	Erro	Tempo	Parada
4	0,018513	0,1823 min	sucesso