



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2015/09.16.22.02-TDI

PREVISÃO CLIMÁTICA DE PRECIPITAÇÃO POR REDES NEURAIS AUTOCONFIGURADAS

Juliana Aparecida Anochi

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Haroldo Fraga de Campos Velho, e Elcio Hideiti Shiguemori, aprovada em 03 de novembro de 2015.

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34P/3K98PDP>>

INPE
São José dos Campos
2015

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

COMISSÃO DO CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELECTUAL DO INPE (DE/DIR-544):

Presidente:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Membros:

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Amauri Silva Montes - Coordenação Engenharia e Tecnologia Espaciais (ETE)

Dr. André de Castro Milone - Coordenação Ciências Espaciais e Atmosféricas
(CEA)

Dr. Joaquim José Barroso de Castro - Centro de Tecnologias Espaciais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos
(CPT)

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Clayton Martins Pereira - Serviço de Informação e Documentação (SID)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Simone Angélica Del Ducca Barbedo - Serviço de Informação e Documentação
(SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Marcelo de Castro Pazos - Serviço de Informação e Documentação (SID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2015/09.16.22.02-TDI

PREVISÃO CLIMÁTICA DE PRECIPITAÇÃO POR REDES NEURAIS AUTOCONFIGURADAS

Juliana Aparecida Anochi

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Haroldo Fraga de Campos Velho, e Elcio Hideiti Shiguemori, aprovada em 03 de novembro de 2015.

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34P/3K98PDP>>

INPE
São José dos Campos
2015

Dados Internacionais de Catalogação na Publicação (CIP)

Anochi, Juliana Aparecida.

An78p Previsão climática de precipitação por redes neurais autoconfiguradas / Juliana Aparecida Anochi. – São José dos Campos : INPE, 2015.

xxiv + 135 p. ; (sid.inpe.br/mtc-m21b/2015/09.16.22.02-TDI)

Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2015.

Orientadores : Drs. Haroldo Fraga de Campos Velho, e Elcio Hideiti Shiguemori.

1. Problemas de otimização. 2. Meta-heurísticas. 3. Rede neural artificial. 4. Previsão climática. 5. Redução de dados.
I.Título.

CDU 519.863:551.515.4



Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](https://creativecommons.org/licenses/by-nc/3.0/).

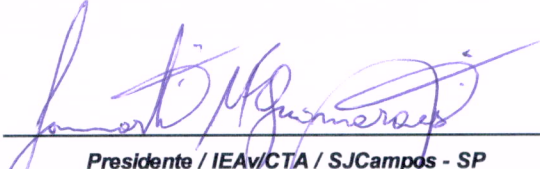
This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](https://creativecommons.org/licenses/by-nc/3.0/).

Aluno (a): **Juliana Aparecida Anochi**

Título: "PREVISÃO CLIMÁTICA DE PRECIPITAÇÃO POR REDES NEURAIAS
AUTOCONFIGURADAS"


Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de **Doutor(a)** em
Computação Aplicada

Dr. Lamartine Nogueira Frutuoso
Guimarães



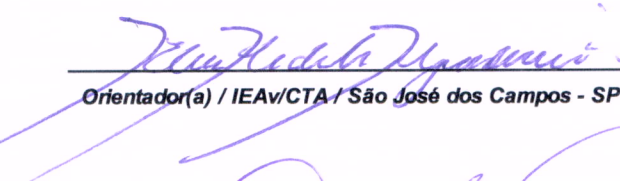
Presidente / IEAv/CTA / SJC Campos - SP

Dr. Haroldo Fraga de Campos Velho



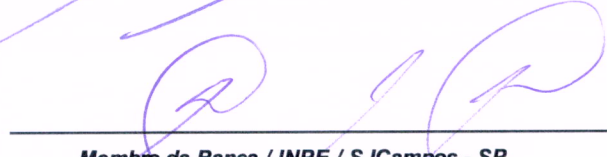
Orientador(a) / INPE / São José dos Campos - SP

Dr. Elcio Hideiti Shiguemori



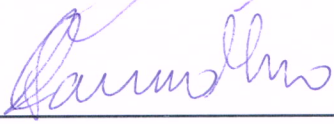
Orientador(a) / IEAv/CTA / São José dos Campos - SP

Dra. Sandra Aparecida Sandri




Membro da Banca / INPE / SJC Campos - SP

Dr. Solon Venâncio de Carvalho



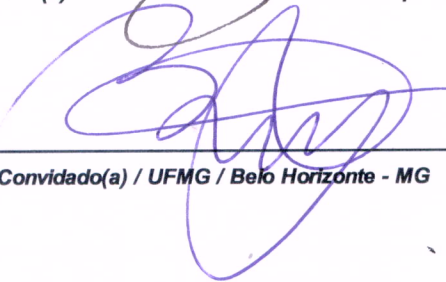
Membro da Banca / INPE / SJC Campos - SP

Dr. Eduardo Fávero Pacheco da Luz



Convidado(a) / CEMADEN / São José dos Campos - SP

Dr. Antonio de Padua Braga



Convidado(a) / UFMG / Belo Horizonte - MG

Este trabalho foi aprovado por:

() maioria simples

unanimidade

São José dos Campos, 03 de novembro de 2015

“ If I have seen further it is by standing on the shoulders of Giants”

Se vi mais longe foi por estar de pé sobre ombros de gigantes

ISAAC NEWTON

*Dedico a meu orientador José Demisio Simões da Silva (in
memoriam)*

AGRADECIMENTOS

Agradeço primeiramente a Deus por mais uma conquista concedida, por me dar força interior para superar as dificuldades. A minha mãe divina, Nossa Senhora Aparecida.

A meus pais, Olimpio José Anochi e Maria Izabel Anochi, pelo constante apoio e pela paciência e atenção que sempre tiveram, pelo entusiasmo e orgulho com que sempre reagiram aos resultados acadêmicos ao longo dos anos. A minhas irmãs, Andreia Aparecida Anochi e Simone Aparecida Anochi pela compreensão de minha ausência em momentos importantes da vida de suas vidas.

Ao meu querido marido Ricardo Miranda Rodrigues, pela compreensão e ternura sempre manifestadas apesar da falta de atenção e das ausências.

Agradeço ao meu orientador professor Dr. Haroldo Fraga de Campos Velho, pela orientação e confiança, por me mostrar o caminho da ciência, por acreditar no futuro deste trabalho e contribuir para o meu crescimento profissional e por ser também um exemplo a ser seguido. Sua participação foi fundamental para a realização deste trabalho. Agradeço ao meu co-orientador Dr. Elcio Hideiti Shiguemori pela orientação na execução deste trabalho.

Um agradecimento especial ao professor Dr. José Demisio Simões da Silva (*in memoriam*), pessoa pela qual tive o privilégio de conhecer, grande mestre na área de redes neurais.

Agradeço a todos os professores e pesquisadores da pós-graduação, com os quais muito aprendi através das disciplinas do curso e seminários, em especial ao Professor Dr. Nelson Jesus Ferreira, que sempre incentivou o meu crescimento profissional.

Agradeço ao Instituto Nacional de Pesquisas Espaciais pela oportunidade em aprimorar a minha qualificação profissional. Ao CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico, pelo apoio financeiro.

Por fim, agradeço a todos os meus amigos de pós-graduação do INPE, em especial Eduardo Luz, Reynier Hernández Torres, Rosângela Cintra, Juliana Marino. À amiga de longa data, Érica Ferreira Souza, com quem posso sempre contar. À amiga Sabrina Sambati, por fazer parte da minha vida nos momentos bons e ruins; por ser minha fiel amiga, sempre fará parte da minha vida. Agradeço de coração por partilharem cada minuto de dificuldade e de alegria nestes anos.

RESUMO

Previsão climática do campo de precipitação é um aspecto chave em meteorologia. Precipitação é uma variável associada a desastres naturais (secas e enchentes) e safras agrícolas, com impactos nos setores de turismo e transporte. Entretanto esta variável meteorológica é de difícil previsão, devido à grande variabilidade temporal e espacial (variável descontínua). Neste trabalho, um método baseado em Rede Neural Artificial (RNA) é aplicado para previsão climática de precipitação nas regiões Sul, Sudeste e Nordeste do Brasil. É conhecida a capacidade de redes neurais de aprendizado e resposta, o que motiva sua aplicação com sucesso em uma grande variedade de problemas, consolidando-se como uma técnica de solução de problemas complexos em reconhecimento de padrões, classificação, sistemas de controle, aproximação de funções e modelo preditivo. Redes neurais podem ser caracterizadas como redes supervisionadas e não supervisionadas. Em geral, o processo de treinamento de redes neurais supervisionadas está associado à determinação dos pesos das conexões. A definição ou identificação da arquitetura ótima para uma rede neural é expressa como um problema de otimização, em que cada ponto no espaço de busca representa uma topologia diferente. O problema de otimização pode ser formulado por meio de uma função mono-objetivo ou de uma função multiobjetivo. Neste trabalho, a otimização mono-objetivo foi solucionada pelo *Multi-Particle Collision Algorithm* (MPCA) e o *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) foi empregado para otimização multiobjetivo. Em meteorologia, dados de diversas fontes (satélites, estações de superfície, boias oceânicas, radiossondagens, radar e muitas outras) são usados nas previsões de tempo e clima. Assim, previsão de eventos meteorológicos é um desafio complexo, mais ainda deve-se incluir a necessidade de análise de grande volume de dados. A redução da dimensão de dados de observação sem perda de informação é um tema importante de pesquisa. A Teoria dos Conjuntos Aproximativos, uma técnica de mineração de dados, foi empregada para identificar as variáveis mais significativas para o processo de previsão climática.

CLIMATE PRECIPITATION PREDICTION BY SELF-CONFIGURED NEURAL NETWORKS

ABSTRACT

Climate precipitation prediction field is a key aspect in meteorology. The precipitation is a variable associated with natural disasters (droughts and floods) agricultural crops and can cause impacts in the sectors of tourism and shipping. However, this is a meteorological variable that is difficult to predict because of large spatial and temporal variability (i.e. variable discontinuous). A method based on Artificial Neural Network (ANN) is applied to climate prediction precipitation in the South, Southeast and Northeast regions of Brazil. It is known the ability of neural network learning and response, which motivates their successful application in a wide variety of problems, consolidating its position as a solution technique of complex problems in pattern recognition, classification, control systems, proximity functions and predictive model. Neural networks can be characterized as supervised and unsupervised networks. In general, the supervised training process for neural networks is associated with the determination of the weights of the connections. The definition or identification of the optimal architecture for a neural network is expressed as an optimization problem, in which each point in the search space represents a different topology. The optimization problem can be formulated by a mono-objective function or a multiobjective function. The mono-objective optimization was solved by Multi-Particle Collision Algorithm (MPCA) and Non-dominated Sorting Genetic Algorithm-II (NSGA-II) was used for multi-objective optimization. In meteorology, data from various sources (satellites, ground-based stations, ocean buoys, soundings, radar and many others) are used in weather and climate forecasts. Predicting meteorological events is a complex challenge. The size reduction of the observation data without losing information is an important subject of research. The Rough Sets Theory, a data mining technique was used to identify the most significant variables for the climate prediction process.

LISTA DE FIGURAS

	<u>Pág.</u>
2.1	Estrutura não-linear de um neurônio artificial. 15
2.2	Funções de ativação. 16
2.3	Arquitetura da Rede Perceptron de Múltiplas Camadas. 18
2.4	Arquitetura da rede recorrente Elman. 19
3.1	Relação de dominância em uma minimização entre duas soluções. 31
3.2	Algoritmo PCA. 35
3.3	Função para perturbação no PCA. 36
3.4	Função para exploração no PCA. 36
3.5	Função para espalhamento no PCA. 37
3.6	Algoritmo de colisão de múltiplas partículas. 38
3.7	Estrutura básica do algoritmo genético. 41
3.8	A 2D Pareto Front 42
3.9	Esquema do algoritmo NSGA-II. 44
4.1	Opções que a ferramenta Rosetta oferece. 54
5.1	MLP-Especialista 68
5.2	MLP-MPCA 68
5.3	MLP-NSGA-II 68
5.4	Erro para as três RNAs 68
6.1	Esquema de previsão sazonal 72
6.2	Área de estudo: Nordeste (NE) 78
6.3	Previsão sazonal de precipitação para a estação Verão 83
6.4	Previsão sazonal de precipitação para a estação Outono 84
6.5	Previsão sazonal de precipitação para a estação Inverno 85
6.6	Previsão sazonal de precipitação para a estação Primavera 86
6.7	Área de estudo: Sul (S) 87
6.8	Previsão sazonal de precipitação para a estação Verão 91
6.9	Mapa de erro para previsão sazonal de precipitação na estação Verão 92
6.10	Previsão sazonal de precipitação para a estação Outono 93
6.11	Mapa de erro para previsão sazonal de precipitação na estação Outono 94
6.12	Resultados de previsão sazonal de precipitação na estação Inverno 95
6.13	Mapa de erro para previsão sazonal de precipitação na estação Inverno 96
6.14	Resultados de previsão sazonal de precipitação para a estação Primavera 97

6.15	Mapa de erro para previsão sazonal de precipitação para a estação Primavera	98
6.16	Área de estudo: Sudeste (SE)	99
6.17	Resultados de previsão sazonal de precipitação para a estação Verão . . .	103
6.18	Mapa de erro para a previsão sazonal de precipitação na estação Verão .	104
6.19	Resultados de previsão sazonal de precipitação para a estação Outono . .	105
6.20	Mapa de erro para a previsão sazonal de precipitação na estação Outono	106
6.21	Resultados de previsão sazonal de precipitação para a estação Inverno . .	107
6.22	Mapa de erro para a previsão sazonal de precipitação na estação Inverno	108
6.23	Resultados de previsão sazonal de precipitação para a estação Primavera	109
6.24	Mapa de erro para a previsão sazonal de precipitação na estação Primavera	110
7.1	Área de estudo: Subárea NE	112
7.2	Previsão mensal de precipitação para o mês de abril – NN:MLP	114
7.3	Previsão mensal de precipitação para o mês de abril – NN:Elman	115
A.1	Exemplo de um projeto no Rosetta	135
A.2	Exemplo de uma área de trabalho do sistema Rosetta	135

LISTA DE TABELAS

	<u>Pág.</u>
4.1 Sistema de Informação.	47
4.2 Sistema de Decisão.	48
4.3 Redução de Informação.	50
4.4 Aproximações do Sistema de Decisão.	51
4.5 Matriz de Discernibilidade.	51
5.1 Parâmetros de controle e critério de parada: MPCA e NSGA-II.	67
5.2 Parâmetros para arquiteturas de RNA	69
6.1 Variáveis selecionadas do NCEP/NCAR	74
6.2 Parâmetros de controle e critério de parada: MPCA e NSGA-II	77
6.3 Parâmetros que definem arquiteturas de RNAs	78
6.4 Resultados para a previsão climática na região Nordeste	79
6.5 Variáveis extraídas pela TCA	80
6.6 Resultados para a previsão climática na região Sul	88
6.7 Variáveis extraídas pela TCA - Sul	88
6.8 Resultados para a previsão climática na região Sudeste	100
6.9 Variáveis extraídas pela TCA - Sudeste	100
7.1 Parâmetros definidos por MPCA	113

LISTA DE ABREVIATURAS E SIGLAS

AG	–	Algoritmo Genético
COLA	–	<i>Center for Ocean Land and Atmosphere Studies</i>
CPTEC	–	Centro de Previsão de Tempo e Estudos Climáticos
CV	–	Coefficiente de variação
FIRMLP	–	<i>Finite Impulse Response Multilayer Perceptron</i>
FK	–	Filtro de Kalman
GrADS	–	<i>Grid Analysis and Display System</i>
IND	–	Relação de Indiscernibilidade
INPE	–	Instituto Nacional de Pesquisas Espaciais
MCGA	–	Modelo de Circulação Geral Atmosférico
MLP	–	<i>Multilayer Perceptron</i>
MPCA	–	<i>Multiple Particle Collision Algorithm</i>
MPI	–	<i>Message Passing Interface</i>
NCAR	–	<i>National Center for Atmospheric Research</i>
NCEP	–	<i>National Centers for Environmental Prediction</i>
NOAA	–	<i>National Oceanic & Atmospheric Administration</i>
NSGA-II	–	<i>Non-dominated Sorting Genetic Algorithm II</i>
PCA	–	<i>Particle Collision Algorithm</i>
PNT	–	Previsão Numérica do Tempo
RNA	–	Rede Neural Artificial
RN	–	Rede Neural
ROSETTA	–	<i>Rough Set Toolkit for Analysis of Data</i>
SA	–	<i>Simulated Annealing</i>
SD	–	Sistema de Decisão
SI	–	Sistema de Informação
SF	–	Sistemas Frontais
TCA	–	Teoria dos Conjuntos Aproximativos
ZCAS	–	Zona de Convergência do Atlântico Sul

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO	1
1.1 Objetivo e Contribuição	5
1.2 Aplicações	6
1.2.1 Assimilação de dados	6
1.2.2 Previsão Climática	7
1.3 Organização do texto	9
2 REDE NEURAL ARTIFICIAL	11
2.1 Breve histórico	11
2.2 Conceitos básicos	13
2.3 Neurônio Biológico	14
2.4 Neurônio Artificial	14
2.5 Função de ativação	15
2.6 Arquitetura de Rede Neural	16
2.6.1 Perceptron de Múltiplas Camadas	17
2.7 Rede recorrente	17
2.7.1 Rede recorrente Elman	18
2.8 Processo de Aprendizagem	19
2.9 Treinamento	20
2.9.1 Treinamento: Algoritmo de retropropagação do erro	21
2.9.2 Treinamento: meta-heurística MPCA	26
3 ALGORITMOS DE OTIMIZAÇÃO	27
3.1 Otimização	27
3.1.1 Otimização mono-objetivo	28
3.1.2 Otimização multiobjetivo	29
3.1.2.1 Formalização do problema de otimização multiobjetivo	29
3.1.2.2 Conjunto e fronteira de Pareto	30
3.2 Algoritmos Meta-heurísticos	32
3.3 Algoritmo de Colisão de Partículas (PCA)	34
3.3.1 Algoritmo de Colisão de Múltiplas Partículas (MPCA)	37
3.4 Algoritmo Genético	39
3.4.1 Operadores genéticos em AG	39

3.4.2	Non-dominated Sorting Genetic Algorithm (NSGA-II)	40
4	TEORIA DOS CONJUNTOS APROXIMATIVOS	45
4.1	Introdução	45
4.2	Sistemas de Informação	46
4.3	Sistemas de Decisão	47
4.4	Relação de Indiscernibilidade	48
4.5	Aproximação de Conjuntos	49
4.6	Matriz de Discernibilidade	51
4.7	Função de Discernibilidade	52
4.8	Redução	52
4.9	Ferramenta Rosetta	53
4.9.1	Pré-processamento	53
4.9.2	Discretização	54
4.9.3	Redução de atributos	55
5	CONFIGURAÇÃO AUTOMÁTICA DE REDES NEURAIIS	57
5.1	Otimização da arquitetura da rede neural por meta-heurística	57
5.1.1	Solução com otimização mono-objetivo por MPCA	58
5.1.1.1	Função Objetivo	58
5.1.1.2	Algoritmo MPCA aplicado em otimização de parâmetros	59
5.1.2	Solução com otimização multiobjetivo com NSGA-II	61
5.1.2.1	Algoritmo NSGA-II aplicado em otimização de parâmetros	61
5.1.3	Testes preliminares com MPCA e NSGA-II	63
5.1.3.1	Filtro de Kalman	65
5.1.3.2	Equação da onda 1D linear	66
5.1.3.3	Resultados preliminares	66
6	REDES NEURAIIS PARA PREVISÃO CLIMÁTICA DE PRECIPITAÇÃO	71
6.1	Previsão Climática	71
6.2	Ambiente de Teste	73
6.2.1	Dados Meteorológicos	73
6.2.2	Visualização de dados	75
6.2.3	Conjunto de dados para treinamento e generalização	75
6.2.4	Redução dos dados meteorológicos com TCA	75
6.3	Resultados com redes autoconfiguradas	76
6.3.1	Previsão Sazonal	78

6.3.1.1	Região Nordeste	78
6.3.1.2	Região Sul	87
6.3.1.3	Região Sudeste	99
7	ESTRATÉGIA DE TREINAMENTO USANDO META- HEURÍSTICA	111
7.1	Método de treinamento usando MPCA	111
7.1.1	Previsão mensal para a região Nordeste	111
7.1.2	Experimentos	112
8	CONCLUSÕES	117
	REFERÊNCIAS BIBLIOGRÁFICAS	121
	APÊNDICE A - SISTEMA ROSETTA	133

1 INTRODUÇÃO

As preocupações causadas pelas mudanças climáticas crescem cada vez mais na comunidade científica. Em meio ao grande destaque que o clima e suas mudanças vêm apresentando na última década, cientistas do mundo inteiro estão procurando compreender a natureza das mudanças que provavelmente ocorrerão, assim como os possíveis impactos que podem ocasionar para a sociedade em geral.

No Brasil, encontra-se uma grande variedade de padrões climáticos. Na região Norte, verifica-se um clima equatorial chuvoso, praticamente sem estação seca. No Nordeste, a estação chuvosa, com baixos índices pluviométricos, restringe-se a poucos meses, caracterizando um clima semi-árido e apresentando alta previsibilidade climática. As regiões Sudeste e Centro-Oeste sofrem influências tanto de sistemas tropicais quanto de latitudes médias, com estação seca bem definida no inverno e estação chuvosa no verão com chuvas convectivas. Ambas regiões apresentam baixa previsibilidade devido à menor dependência em relação as condições dos oceanos e à grande variedade dos sistemas meteorológicos que as atingem. Finalmente, a região Sul do Brasil é caracterizada com média previsibilidade, e devido à sua localização latitudinal, sofre mais influência dos sistemas de latitudes médias, onde os sistemas frontais são os principais causadores de chuvas durante o ano (SAMPAIO; SILVA DIAS, 2014).

Na região Sudeste, uma crise hídrica iniciou-se em fevereiro de 2014. O estado de São Paulo, por exemplo, sofreu uma grande seca em 2014 e início de 2015, levando a uma série de impactos em diversos setores sócio-econômicos. Reservatórios em torno da região metropolitana de São Paulo atingiram níveis baixos sem precedentes. O principal manancial (Sistema Cantareira) que normalmente atende a maior parte da população da Grande São Paulo, chegou em janeiro de 2015 ao nível de apenas 5% da capacidade de armazenamento. A seca prolongada teve impactos não só na disponibilidade de água para consumo público, mas também para a produção agrícola e geração de energia hidrelétrica (COELHO et al., 2015).

Essa dependência dos fatores climáticos motivou o desenvolvimento deste trabalho, cujo objetivo é realizar a previsão climática do campo de precipitação para as regiões Sul, Sudeste e Nordeste do Brasil, utilizando redes neurais artificiais como modelos preditivos.

Redes neurais artificiais (RNA) são métodos computacionais cujo princípio de funcionamento é regido por um modelo matemático inspirado no funcionamento dos elementos básicos que formam a estrutura neural de organismos inteligentes, que

adquirem conhecimento através de experiência. Seu comportamento resulta das interações entre as unidades de processamento, a partir de seu ambiente, através de um processo de aprendizagem (HAYKIN, 1998). As redes neurais são como excelentes ferramentas na derivação de modelos de dados, devido à sua característica inerente de plasticidade que permite a adaptação da tarefa de aprendizagem quando os dados são fornecidos.

O interesse na utilização em rede neural vem de sua capacidade de aprendizado. Sua popularidade é atribuída ao fato de ter sido aplicada com sucesso a uma grande variedade de problemas, consolidando-se como uma técnica de solução de problemas complexos em reconhecimento de padrões, classificação de padrões, sistemas de controle, aproximação de funções e modelo preditivo.

Apesar do potencial de um modelo de rede neural, seu desempenho é dependente da definição dos seus parâmetros, visto que, a definição da arquitetura (topologia) pode influenciar significativamente no processo de treinamento.

A definição de parâmetros da arquitetura de uma RNA é um problema complexo e geralmente requer um considerável esforço por parte do especialista na determinação de alguns parâmetros. A abordagem mais usada é uma avaliação empírica, em que várias configurações de redes são avaliadas pelo especialista, isto é, os parâmetros livres da rede neural artificial são modificados. O objetivo é identificar uma configuração que permita resolver satisfatoriamente o problema em estudo. Entretanto, essa abordagem além de exigir do especialista o conhecimento teórico de RNA, também lhe é exigido o conhecimento prévio sobre aplicação, de tal forma que os ajustes dos parâmetros possam gerar um modelo que represente bem o problema.

Embora o método empírico seja popular, nem sempre apresenta resultados autênticos, uma vez que não há confiabilidade na obtenção de uma arquitetura ótima (PRECHELT et al., 1994).

A definição da arquitetura das redes neurais supervisionadas do tipo Perceptron de Múltiplas Camadas (MLP) e rede recorrente Elman, utilizadas neste trabalho, consiste basicamente na determinação dos parâmetros: número de camadas escondidas (intermediária), número de neurônios em cada camada, tipo da função de ativação, taxa de aprendizagem e taxa *momentum*. A quantidade de neurônios na camada intermediária é crucial no processo de treinamento. Uma quantidade pequena de neurônios pode ser insuficiente para a rede neural aprender os padrões existentes nos dados (*underfitting*). Por outro lado, quanto maior a quantidade de neurônios

na camada intermediária, maior é o espaço de busca de soluções encontradas no processo de treinamento, o que pode conduzir à perda de capacidade de generalização (*overfitting*).

Existem na literatura vários métodos para o treinamento de redes neurais supervisionadas que visam minimizar os efeitos provocados pelo *underfitting* e *overfitting*. Técnicas como a utilização de métodos de validação, algoritmos construtivos, algoritmos de poda e regularização, têm sido aplicadas visando a melhoria da capacidade de generalização. Nas abordagens por validação, como o algoritmo de validação cruzada (*cross validation*) proposto por Stone (1978), uma medida de erro sobre um conjunto de dados de validação (teste) é usada para estimar os parâmetros durante a fase de treinamento. Uma interrupção precoce no treinamento é realizada quando o erro dos padrões do conjunto de validação cresce. Algoritmos de poda ou construtivos, são baseados na manipulação da arquitetura da rede, que na prática resultam em modelos menos complexos apresentando um número menor de parâmetros, como por exemplo, o algoritmo de poda (*pruning*) introduzido por Reed (1993). A utilização de técnica de regularização, como o conhecido algoritmo de decaimento de peso (*weight decay*) proposto por Weigend e Huberman (1990), controla a complexidade dos modelos de RNA, a partir de restrições na magnitude dos pesos sinápticos da rede.

Neste contexto, fica evidente que a busca de uma arquitetura ótima para uma RNA é um tema de pesquisa muito relevante. Nas últimas décadas, novos algoritmos de aprendizado para modelos de redes neurais, baseados em técnicas de otimização mono-objetivo e multiobjetivo, têm atraído grande atenção da comunidade científica. Nessa abordagem, os parâmetros e pesos são definidos por um processo de otimização, em que o desempenho da rede neural corresponde à minimização (ou maximização) de uma determinada função objetivo.

Teixeira et al. (2000) apresentam um algoritmo de otimização multiobjetivo visando melhorar a capacidade de generalização e controlar a complexidade do modelo MLP, através da minimização de dois objetivos conflitantes: o erro de treinamento e a norma do vetor de pesos.

Rocha (2012) apresenta duas abordagens de otimização multiobjetivo, uma para o treinamento da rede neural e outra para seleção de características. Para o método de treinamento, o algoritmo de evolução diferencial foi utilizado para a modificação dos pesos sinápticos da rede, minimizando o erro de treinamento e a norma do vetor de pesos. Para a seleção de características, o autor utilizou o algoritmo de seleção

clonal como estratégia para a realização de busca em um processo de seleção, visando reduzir a quantidade de atributos do problema.

Costa et al. (2003a) utilizaram um método de otimização multiobjetivo para garantir uma boa capacidade de generalização e para simplificar a arquitetura de uma rede MLP, por meio da minimização de duas funções objetivos: erro de treinamento e a norma dos vetores de pesos. O método proposto foi um algoritmo de poda construtivo.

Costa et al. (2003b) apresentam a otimização do treinamento de RNAs tratado como problema multiobjetivo, usando o algoritmo de controle por modos deslizantes (*Sliding Mode Control*) formado por duas funções objetivo: o erro de treinamento e a norma dos vetores de peso. Este algoritmo controla a convergência da rede para o ponto de mínimos de forma que as superfícies de deslizamento sejam alcançadas.

Benardos e Vosniakos (2007) apresentam um problema de otimização multiobjetivo para a definição de parâmetros de uma arquitetura de RNA. Um algoritmo genético (GA) é empregado para buscar no espaço de soluções e a melhor arquitetura é determinada a partir de um conjunto de critérios pré-definidos. Os resultados mostram que a abordagem possui um melhor desempenho comparado com o método empírico.

Carvalho et al. (2011) desenvolveram uma abordagem para otimizar arquiteturas de RNA do tipo MLP, utilizando técnicas de otimização mono-objetivo. Para a resolução deste problema, representado pela busca ótima de uma arquitetura, os autores utilizaram quatro meta-heurísticas de busca global: otimização extrema generalizada, *variable neighborhood search*, *simulated annealing* e algoritmo genético.

O problema de definição de parâmetros de uma arquitetura ótima para uma rede neural pode ser formulado como um problema de otimização, em que cada ponto no espaço de busca representa uma arquitetura diferente. O objetivo é encontrar um valor ótimo, a partir de uma função objetivo, a qual pode levar em conta, por exemplo, o erro de treinamento e a complexidade da rede (número de neurônios). O custo de todas as arquiteturas, formam uma superfície discreta de busca, de modo que a definição dos parâmetros ótimos, consiste em topologias que apresentem um menor erro, associado a uma rede de baixa complexidade. Ou seja, uma configuração que minimize a função objetivo (CARVALHO et al., 2011).

Neste trabalho, para o problema de otimização de parâmetros utilizou-se duas estratégias distintas: um problema de minimização mono-objetivo utilizando o Algoritmo

de Colisão de Múltiplas Partículas (em inglês *Multi-Particle Collision Algorithm - MPCA*) introduzido por Luz et al. (2008) e um problema de minimização multi-objetivo utilizando o Algoritmo Genético de Ordenação não-dominado (em inglês *Non-Dominated Sorting Genetic Algorithm - NSGA-II*) proposto por Deb et al. (2002).

Em meteorologia, dados de diversas fontes são usados nas previsões de tempo e clima. Assim, previsão de eventos meteorológicos é um desafio complexo, mais ainda deve-se incluir a necessidade de análise de grande volume de dados. Identificar, com razoável antecedência e confiabilidade, condições favoráveis à ocorrência de eventos climáticos e realizar em tempo hábil para que se possa tomar as devidas medidas e conseqüentemente reduzir o impacto negativo de eventos meteorológicos, é algo essencial para a sociedade. A redução da dimensão de dados de observação sem perda de informação é um tema importante de pesquisa.

Anochi (2010) propõe o uso de técnicas de mineração de dados como um método de redução de dimensionalidade de observações, para o processo de previsão climática de precipitação, usando a TCA para extrair informações relevantes da base de dados. Os dados reduzidos foram usados como entrada para um modelo de previsão baseado em redes neurais. Esta mesma metodologia é empregada na presente tese.

Neste contexto, outro tópico abordado neste trabalho é o uso da Teoria dos Conjuntos Aproximativos (TCA) como técnica de redução de dimensão de dados meteorológicos.

1.1 Objetivo e Contribuição

Como visto, a determinação de uma configuração apropriada para redes neurais é uma tarefa complexa, que em geral requer o conhecimento de especialistas sobre a aplicação e também em redes neurais. A presente pesquisa faz uso de técnicas de otimização da configuração de redes neurais supervisionadas. Buscando determinar a arquitetura ideal da rede de forma automática. Esta dificuldade é pouco discutida, ou até mesmo totalmente ignorada, em grande parte dos trabalhos.

Os objetivos específicos são:

- Desenvolver um modelo preditivo de previsão climática do campo de precipitação em mesoescala;
- Usar rede neural artificial como uma técnica não convencional para assi-

milação de dados;

- Utilizar meta-heurísticas para o treinamento automático das redes neurais MLP e Elman;
- Configurar automaticamente arquiteturas de redes neurais supervisionadas usando os algoritmos de otimização MPCA e NSGA-II, sem necessitar da experiência de um especialista na determinação de parâmetros e o problema em estudo;
- Reduzir a dimensionalidade dos dados meteorológicos utilizando a TCA;
- Avaliar a abordagem proposta aplicada em dois estudos de casos: assimilação de dados e previsão climática.

1.2 Aplicações

Na presente pesquisa, redes neurais são usadas como modelos alternativos em duas aplicações da meteorologia: uma técnica não convencional para assimilação de dados e como modelo preditivo de previsão climática de precipitação.

1.2.1 Assimilação de dados

Os modelos numéricos de previsão de tempo e clima necessitam de dados observados que sirvam como condição inicial para sua execução. Tais dados, são fornecidos por estações terrestres de observações de condições meteorológicas que formam a rede de observações distribuídas pelo planeta para coleta de dados.

Uma importante linha de pesquisa em meteorologia, denominada assimilação de dados, consiste em estimar o estado de um sistema dinâmico por meio de observações, aprimorando as condições iniciais do modelo numérico de previsão de tempo e, conseqüentemente, a previsão fornecida por este modelo.

A evolução histórica dos métodos de assimilação de dados passa pelo Ajuste de Funções, Correções Sucessivas, Análise de Correções, Interpolação Ótima, Métodos Variacionais, Filtro de Kalman e Técnicas de Monte Carlo (DALEY, 1993; KALNAY, 2003; GRIFFITH, 1997). Todos esses métodos implicam em diferentes técnicas em que combinam uma previsão de “*background*” (estimativa *a priori* do modelo de previsão) com observações, resultando no dado de análise ou condição inicial, o qual será usado para iniciar os modelos numéricos para o ciclo seguinte de previsão.

Os componentes básicos para sistemas operacionais de previsão são: a rede de dados de observação, o modelo numérico e o método de assimilação de dados. As técnicas de assimilação de dados são uma alternativa para combinar de modo eficaz os dados observados com os dados do modelo físico matemático. Essas técnicas podem ser baseadas em teoria da estimação (como: Filtro de Kalman (FK) (KALMAN, 1960); Filtro de Partículas (FP) (GORDON et al., 1993)); no cálculo variacional (TALAGRAND; COURTIER, 1987) (como: método do *Representante* (BENNETT, 2002)), ou ainda, baseadas em Inteligência Artificial (IA), com o uso de Redes Neurais Artificiais (NOWOSAD et al., 2000), (HÄRTER; CAMPOS VELHO, 2005), (CINTRA, 2010).

Cintra e CAMPOS VELHO (2014) utilizaram a rede MLP para emular os resultados da técnica de Filtro de Kalman por Conjunto Transformado e Localizado (*Local Transform Ensemble Kalman Filter* - LETKF). Este esquema é uma aproximação do FK, em que um conjunto tipo Monte-Carlo de previsões de curto prazo são usadas para estimar as covariâncias do erro do modelo de previsão. Os autores mostraram que a principal vantagem do uso de redes neurais em assimilação de dados foi a velocidade computacional na obtenção da condição inicial do modelo que acelera todo o processo de previsão numérica do tempo.

Furtado et al. (2011) apresentam uma técnica de assimilação de dados que utiliza a rede MLP aplicada ao modelo de água rasa em duas dimensões e na equação da onda em uma dimensão. A rede neural foi aplicada para emular o Filtro de Kalman e o método Representante. Os autores mostraram que o método empregado foi eficiente, alcançando a mesma qualidade das análises obtidas por meio do FK e o método Representante e possivelmente uma alternativa viável para o problema de assimilação de dados.

Neste trabalho, foram utilizados dois modelos de redes neurais, os quais foram configurados automaticamente por meio de algoritmos de meta-heurística: a rede MLP e a rede recorrente Elman. Ambas as redes foram treinadas para emular o Filtro de Kalman, utilizando a equação da onda de primeira ordem.

1.2.2 Previsão Climática

O segundo estudo de caso proposto neste trabalho, foi o desenvolvimento de um modelo empírico de previsão climática do campo de precipitação, para as regiões Sul, Sudeste e Nordeste, a partir de dados históricos de reanálise, da base de dados do *National Oceanic & Atmospheric Administration* (NOAA).

Dentre as diversas saídas fornecidas pelos modelos de previsão de tempo e clima, a precipitação é uma das variáveis meteorológicas de maior interesse para a sociedade, tendo em vista que sua ocorrência é um fator limitante para a execução de algumas atividades humanas, especialmente, aquelas ligadas à agricultura, geração de energia hidroelétrica, transportes e, mais recentemente, defesa civil.

A previsão climática é um conjunto de condições naturais que dominam uma determinada região, obtidas pelo comportamento médio da atmosfera, tais como, as variações e as condições extremas do tempo em um intervalo de tempo maior, que são essenciais para caracterizar uma região.

Sousa e Sousa (2010) propuseram um modelo baseado em redes neurais para simular e prever vazões médias mensais na estação fluviométrica localizada na cidade de Piancó, no semiárido paraibano. O modelo foi elaborado a partir da elaboração de várias arquiteturas de redes neurais com base em dados médios de precipitação mensal.

Dornelles (2007) mostrou que os sistemas de alerta de cheias exigem dos modelos de previsão de níveis, precisão e antecipação adequadas e, especialmente, em bacias pequenas com resposta rápida. Estas necessidades são atendidas com modelos de previsão contínua, e que utilizam a previsão hidrometeorológica como dado de entrada. O autor propõe o uso de uma rede neural como modelo de previsão de níveis e, tem como entradas, níveis observados em dois dias anteriores e previsão numérica de precipitação do modelo ETA-CPTEC, para até cinco dias à frente.

Baboo e Shereef (2010) propuseram o uso de uma rede neural como modelo de previsão da variável temperatura. Os autores mostraram que a rede neural é uma poderosa ferramenta capaz de capturar e representar as relações complexas entre muitos fatores que contribuem para a temperatura e, eficiente, como um modelo de predição para a temperatura. Os autores mostraram que o uso de rede neural apresentou bons resultados e pode ser considerada como uma alternativa às abordagens tradicionais meteorológicas, capaz de determinar a relação não linear existente entre os dados históricos (temperatura, velocidade do vento, umidade) fornecidos ao sistema durante a fase de treinamento.

Ruivo et al. (2015) utilizaram metodologias de mineração de dados para investigar as causas de eventos climáticos extremos. Os autores propõem duas metodologias: classificação estatística e árvores de decisão, aplicadas sucessivamente, a fim de reduzir a complexidade do conjunto de dados original. A metodologia foi testada para

análise das grandes secas no Amazonas, ocorridas em 2005 e 2010, e na precipitação extrema ocorrida em Santa Catarina em 2008, em que foi possível apontar alguns parâmetros climatológicos responsáveis por tais eventos.

Neste mesmo tema, [Anochi \(2010\)](#) utilizou técnicas de mineração de dados como um método de redução de atributos, para realização de previsão climática de precipitação por redes neurais. Para o desenvolvimento desta metodologia utilizou-se a Teoria dos Conjuntos Aproximativos para extrair informações relevantes dos dados, visando reduzir a redundância entre as variáveis meteorológicas. O processo de previsão climática de precipitação foi desenvolvido sobre todo território Brasileiro.

Dentro deste contexto, o modelo de previsão desenvolvido neste trabalho, proporciona a elaboração de cenários futuros que apoiam os estudos de impactos e vulnerabilidade e ainda podem permitir a elaboração de projeções dos extremos climáticos do estado atmosférico.

Devida a sua alta variabilidade espacial e temporal, a precipitação impõe um grande desafio à previsão climática e de tempo. Novas técnicas podem e são empregadas para estimar a variável de precipitação. Neste trabalho, a previsão do campo de precipitação, foi realizada usando as redes supervisionadas MLP e a rede recorrente Elman.

1.3 Organização do texto

O texto aqui apresentado está organizado da seguinte forma:

- O [Capítulo 2](#) apresenta de forma sucinta os principais conceitos relacionados às Redes Neurais Artificiais, especialmente, os modelos Perceptron de Múltiplas Camadas e a rede recorrente Elman, os quais foram utilizados neste trabalho.
- O [Capítulo 3](#) são abordados conceitos relativos à otimização mono-objetivo e multiobjetivo, necessários à compreensão dos algoritmos propostos.
- O [Capítulo 4](#) descreve os principais conceitos de Teoria dos Conjuntos Aproximativos, um paradigma para tratamento de incertezas.
- O [Capítulo 5](#) detalha o treinamento automático de redes neurais utilizando algoritmos de otimização mono-objetivo e multiobjetivo. O capítulo também apresenta resultados preliminares obtidos após testes e simulações realizadas com técnicas de otimização.

- O [Capítulo 6](#) apresenta os resultados obtidos após os experimentos realizados com a abordagem de otimização de parâmetros utilizando as meta-heurísticas MPCA e NSGA-II.
- O [Capítulo 7](#) apresenta as conclusões e perspectivas futuras para o trabalho.

2 REDE NEURAL ARTIFICIAL

Neste capítulo é apresentada uma breve introdução às Redes Neurais Artificiais. São discutidos os conceitos básicos sobre o neurônio biológico e artificial, assim como o funcionamento das redes neurais e, principalmente, as fases de treinamento e ativação. As descrições das redes do tipo Perceptron de Múltiplas Camadas (MLP) e rede recorrente Elman, utilizadas neste trabalho são apresentadas.

2.1 Breve histórico

As primeiras pesquisas para o desenvolvimento de computadores baseados no comportamento no cérebro humano datam em 1943, quando McCulloch e Pitts (1943) publicaram o primeiro estudo descrevendo as redes neurais, unificando os estudos de neurofisiologia e lógica matemática e apresentaram um modelo matemático baseado nos neurônios biológicos (MCCULLOCH; PITTS, 1943). Após quatro anos, Pitts e McCulloch (1947) exploraram paradigmas de rede neural para o reconhecimento de padrões usando um perceptron de camada simples.

Em 1949, a publicação do livro *The Organization of Behavior* de Hebb (1949) foi outro marco significativo das redes neurais. Foi apresentada pela primeira vez uma formulação de uma regra de aprendizagem para modificação da sináptica dos neurônios biológicos. Hebb (1949) propôs que a conectividade do cérebro é continuamente modificada conforme um organismo vai aprendendo tarefas funcionais e apresentou o processo de aprendizagem, que determina que a eficiência de uma sinapse variável entre dois neurônios, é aumentada pela ativação e quanto mais as suas saídas estiverem correlacionadas, maior será a intensidade da ligação entre eles (HAYKIN, 1998).

No período entre 1951 e 1958, pouco se evoluiu nas pesquisas de redes neurais. Entretanto, em 1958, Rosenblatt (1958) propõe o *Perceptron* combinando o modelo proposto por (MCCULLOCH; PITTS, 1943) com as percepções biológicas para produzir a primeira rede neural artificial para classificar padrões linearmente separáveis, usando aprendizagem supervisionada.

Em 1960, Widrow e Hoff (MINSKY; PAPER, 1969) conceberam o algoritmo do mínimo quadrado médio (LMS, *least mean-square*) e o utilizaram para formular o *Adaline* (*Adaptive Linear Element*, elemento linear adaptativo). Widrow (1962) apresentou a *Madaline* (*multiple-adaline*), uma das primeiras redes neurais em camadas treinadas com múltiplos elementos adaptativos.

Em 1969, surgiu o livro dos pesquisadores [Minsky e Papert \(1969\)](#), que apresentaram críticas ao *perceptron*, utilizando a matemática para demonstrar que existem limites fundamentais para os perceptrons de uma única camada. Em repercussão as críticas de [Minsky e Papert \(1969\)](#), as pesquisas na área de redes neurais artificiais foram descontinuadas.

O reaquecimento das pesquisas surgiu em 1982 com a publicação do artigo de [Hopfield \(1982\)](#), no qual apresentou um modelo não linear conhecido como rede de Hopfield. Hopfield enfatiza as propriedades associativas e concebeu o conceito de função de energia às redes, com aprendizagem não supervisionada.

Neste mesmo ano, com a publicação de [Kohonen \(1982\)](#), surgiu outro desenvolvimento sobre o aprendizado competitivo com os mapas auto-organizáveis, utilizando uma estrutura de rede unidimensional ou bidimensional, tornando-se conhecido como mapas de atributos auto-organizáveis SOFM (*self-organization feature maps*).

Na década de 80, ressurgiram as redes baseadas em *perceptron* com o modelo de rede *perceptron de múltiplas camadas*, usando um algoritmo chamado *backpropagation* ou retropropagação, com aprendizagem supervisionada, que resolveu em grande parte os problemas levantados por [Minsky e Papert \(1969\)](#).

[Rummelhart et al. \(1986\)](#) conseguiram resolver uma grande parte dos problemas de aprendizagem existentes, como a utilização do algoritmo para aprendizagem de máquina e demonstraram como funcionava. Desenvolveram o algoritmo de retropropagação (*backpropagation*). Naquele mesmo ano, foi publicado o livro *Parallel Distributed Processing: Explorations in the Microstructures of Cognition* editado por [Rumelhart e McClelland \(1986\)](#). Este livro exerceu uma grande influência na utilização da aprendizagem por retropropagação, que emergiu como o algoritmo de aprendizagem mais conhecido e utilizado no treinamento de perceptrons de múltiplas camadas. Este algoritmo foi aplicado com sucesso em uma grande variedade de problemas, o que estimulou o desenvolvimento de muitas pesquisas em redes neurais artificiais.

O algoritmo de aprendizagem *backpropagation* foi criado na mesma época por diferentes pesquisadores. Em 1974, [Werbos \(1974\)](#) descreve em sua tese de doutorado, sobre a computação eficiente do gradiente em modo reverso aplicada em modelos de redes neurais; Em 1985, os pesquisadores [Parker \(1985\)](#), [LeCun \(1985\)](#), também utilizaram o algoritmo *backpropagation*. Entretanto, deve-se atribuir o crédito pelo algoritmo de retropropagação a [Rummelhart et al. \(1986\)](#) que mostraram a utili-

zação para aprendizagem de máquina e como poderia funcionar. Detalhes sobre as notas históricas da era de redes neurais podem ser encontradas em [Haykin \(1998\)](#).

2.2 Conceitos básicos

Redes Neurais Artificiais (RNAs) são modelos computacionais inspirados na estrutura neural de organismos inteligentes. Seu comportamento inteligente, surge das interações entre as unidades de processamento, a partir de seu ambiente, através de um processo de aprendizagem, cuja função é alterar os pesos sinápticos da rede e, disponibiliza este conhecimento para a aplicação em questão.

Redes neurais são sistemas paralelos distribuídos, compostos por neurônios ou unidades de processamento, que computam determinadas funções matemáticas (normalmente não-lineares). Tais neurônios de processamento podem ser distribuídos em uma ou mais camadas e interligados por um grande número de conexões (pesos sinápticos), os quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede.

O interesse em redes neurais vêm de sua capacidade de aprender e responder. Como resultado, as redes neurais têm sido utilizadas em um grande número de aplicações complexas e provaram ser eficazes em uma variedade de campos. Estes incluem o reconhecimento de padrões, classificação, visão, sistemas de controle e predição. Este interesse renovado é refletido no número de cientistas, quantidade de fundos, número de grandes conferências realizadas, e no número de revistas científicas associados com redes neurais ([HAYKIN, 1998](#); [FAUSETT, 1994](#); [BRAGA et al., 2000](#)).

Outras características dos modelos neurais são: não-linearidade, mapeamento de entrada-saída, adaptabilidade, aprendizagem com exemplos, processamento paralelo e distribuído, capacidade de generalização, tolerância à falhas e baixo custo computacional (após o treinamento) ([HAYKIN, 1998](#)).

As redes neurais se assemelham na estrutura do cérebro em dois aspectos:

- Adquire conhecimento através do processo de aprendizagem, o qual, envolve a capacidade de aprender através de exemplos, onde os pesos sinápticos são modificados, de forma ordenada para alcançar a resposta desejada.
- Armazena o conhecimento obtido no processo de aprendizagem, através das conexões sinápticas.

2.3 Neurônio Biológico

O cérebro humano possui cerca de 100 bilhões de neurônios biológicos, sua célula fundamental. Cada um destes neurônios processa e se comunica com milhares de outros continuamente e em paralelo (BRAGA et al., 2000).

Os neurônios possuem um papel essencial na determinação do funcionamento e comportamento do corpo humano, os quais são divididos em três seções: o corpo da célula; os dendritos, que são um conjunto de terminais de entrada e; pelos axônios que são longos terminais de saída.

O neurônio biológico é caracterizado como um dispositivo computacional elementar do sistema nervoso, em que possui muitas entradas e uma saída. As entradas são formadas através das conexões sinápticas que conectam os dendritos aos axônios de outras células nervosas. Os sinais que chegam por estes axônios são pulsos elétricos conhecidos como impulsos nervosos e constituem a informação que o neurônio processa para produzir como saída um impulso nervoso no seu axônio (KOVACS, 2002).

2.4 Neurônio Artificial

O modelo de neurônio artificial proposto por McCulloch e Pitts (1943), interpreta o funcionamento do neurônio biológico como um circuito binário simples que combina várias entradas e apenas um sinal de saída. Sua descrição matemática resultou em um modelo com n terminais de entrada representando os dendritos, e apenas uma saída simulando o axônio. Para emular o comportamento das sinapses, os terminais de entrada do neurônio artificial possuem pesos. Em termos matemáticos, podemos descrever um neurônio k escrevendo o seguinte par de equações (HAYKIN, 1998):

$$v_k = \sum_{j=1}^m w_{kj} x_j, \quad (2.1)$$

e

$$y_k = \varphi(v_k + b_k), \quad (2.2)$$

em que x_1, x_2, \dots, x_m são os sinais de entrada; $w_{k1}, w_{k2}, \dots, w_{km}$ são os pesos sinápticos do neurônio k ; v_k é a saída do combinador linear devido aos sinais de entrada; b_k é o bias; φ é a função de ativação; e y_k é o sinal de saída do neurônio. Uma descrição do modelo artificial de um neurônio está ilustrada na Figura 2.1.

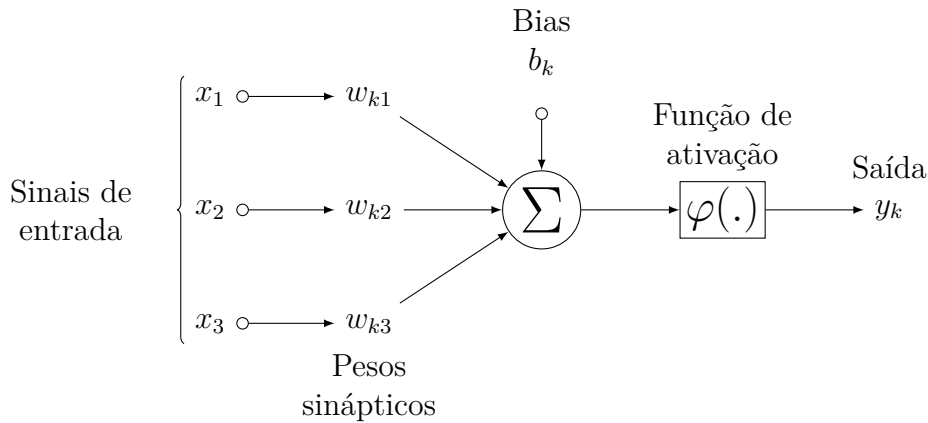


Figura 2.1 - Estrutura não-linear de um neurônio artificial.

Fonte: Adaptada de Haykin (1998).

2.5 Função de ativação

A partir do modelo proposto por McCulloch e Pitts (1943) foram derivados vários outros modelos que permitem a produção de uma saída qualquer e com diferentes funções de ativação.

A função básica do neurônio artificial, ou elemento processador, é realizar o somatório, ponderado por pesos sinápticos, dos elementos do vetor de entrada e aplicar este resultado como entrada de uma função não linear denominada função de ativação.

A função de ativação, representada por φ restringe a amplitude do sinal na saída de um neurônio em termos do campo local induzido v . Existem diversas funções de ativação que podem ser aplicadas para gerar neurônios distintos. A Figura 2.2 ilustra, graficamente, duas principais funções de ativação, a função logística sigmoide e a função tangente hiperbólica.

A função logística sigmoide apresentada na Figura 2.2 apresenta propriedades assintóticas e de suavidade definida como:

$$\varphi(v) = \frac{1}{1 + e^{-av}}, \quad (2.3)$$

em que a é o parâmetro de inclinação da função sigmoide e v é o valor da função de ativação do neurônio. A função logística é do tipo unipolar, onde a saída do modelo varia entre $[0,1]$.

A função tangente hiperbólica [Figura 2.2](#) é uma função sigmoide do tipo bipolar, a qual permite valores positivos e negativos, que varia entre $[-1,1]$ é expressa como:

$$\varphi(v) = \frac{1 - e^{-av}}{1 + e^{av}}. \quad (2.4)$$

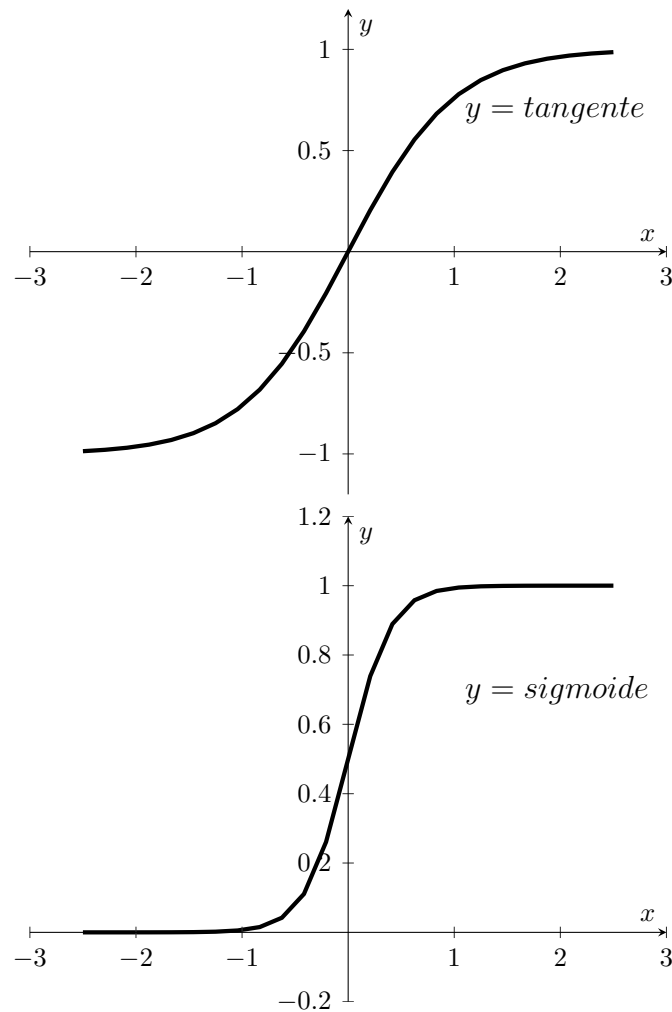


Figura 2.2 - Funções de ativação.

2.6 Arquitetura de Rede Neural

As diversas arquiteturas de redes neurais podem ser formadas pela combinação de neurônios artificiais e são definidas pelo tipo de conexão entre as redes. Cada neurônio transfere o seu sinal apenas para os neurônios que se encontram em uma das camadas subsequentes. [Haykin \(1998\)](#) relata três tipos de camadas.

- Camada de entrada: onde os padrões são apresentados à rede.
- Camadas intermediárias: trabalham como um reconhecedor de características que ficam armazenadas nos pesos sinápticos e são responsáveis pela maior parte do processamento e não tem ligação direta com o usuário externo.
- Camada de saída: onde os sinais de saída da rede são apresentados.

2.6.1 Perceptron de Múltiplas Camadas

A rede Perceptron de Múltiplas Camadas surgiu como alternativa para solução de problemas não linearmente separáveis e tem sido aplicada com sucesso para solução de diversos problemas complexos, através do seu treinamento supervisionado usando o algoritmo de retropropagação de erro (*error backpropagation*) baseado na regra de aprendizagem por correção do erro (HAYKIN, 1998).

Os sinais de entrada são propagados camada por camada, até que o vetor de saída seja obtido na última camada. O processo de aprendizado termina quando o erro desejado é atingido ou quando se atinge um número máximo de épocas de aprendizado (HAYKIN, 1998).

A arquitetura de uma rede MLP consiste no arranjo topológico das unidades de processamento dos neurônios com os respectivos valores de pesos associados às conexões. Os pesos sinápticos são ajustados para fazer com que a resposta real da rede se aproxima da resposta desejada (HAYKIN, 1998). A Figura 2.3 apresenta um exemplo de arquitetura de uma rede MLP, a qual possui uma camada de entrada, uma camada intermediária (oculta) e uma camada de saída.

2.7 Rede recorrente

Redes recorrentes são definidas como aquelas que possuem conexões de realimentação que proporcionam um comportamento dinâmico. Existem dois tipos de redes recorrentes: aquelas em que o padrão de entrada é fixo e a saída caminha, dinamicamente, para um estado estável e aquelas em que ambas, entrada e saída, variam com o tempo, sendo estas últimas utilizadas com maior frequência.

As redes recorrentes ampliam a potencialidade de modelagem em dados temporais, e são utilizadas com sucesso em processamento de dados temporais, reconhecimento de padrões, e em previsões de séries temporais.

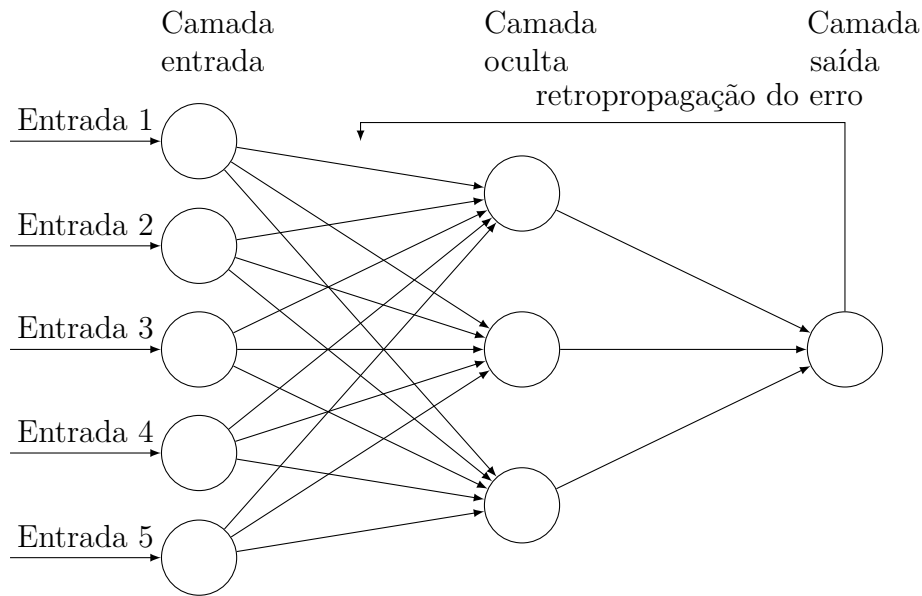


Figura 2.3 - Arquitetura da Rede Perceptron de Múltiplas Camadas.

Para uma rede neural ser considerada dinâmica é necessário possuir memória (ELMAN, 1990). Existem duas maneiras de fornecer memória a uma rede neural. A primeira forma é utilizar atraso no tempo, tais como as técnicas de *Time Delay Neural Network* (TDNN) e *Finite Impulse Response Multilayer Perceptron* (FIRMLP). A segunda maneira, consiste na utilização de redes neurais recorrentes, tais como *Backpropagation Through Time*, *Real-Time Recurrent Learning*, *Cascade Correlation recorrente*, redes Elman e Jordan (BRAGA et al., 2000).

2.7.1 Rede recorrente Elman

A arquitetura da rede de Elman incorpora um perceptron de múltiplas camadas e explora a capacidade de mapeamento não-linear da MLP. Contém conexões recorrentes dos neurônios ocultos para uma nova camada de entrada com neurônios adicionais, chamada de unidade (camada) de contexto, que consiste em unidades de atrasos. Estas unidades de contexto são a memória da rede usada para armazenar o valor de saída das camadas intermediárias por um passo de tempo, e as realimenta de volta para a camada de entrada (ELMAN, 1990).

A rede de Elman é considerada parcialmente recorrente, pois as conexões *feedforward* são modificáveis, e as conexões recorrentes são fixas. A recorrência ocorre através da unidade de contexto, em que possui um atraso do tipo $n - 1$, o qual armazena a

saída da primeira camada oculta para um passo de tempo.

A Figura 2.4 apresenta um exemplo de arquitetura da rede recorrente de Elman, a qual possui quatro camadas: uma camada de entrada composta por neurônios de armazenamento que recebem um sinal externo e propagam sem modificá-los; uma camada intermediária, uma camada de contexto, utilizada para memorizar as ativações dos neurônios da camada intermediária, operando como atrasadores em um instante de tempo; e uma camada saída. A realimentação ocorre da saída de cada neurônio da camada escondida para as unidades de contexto. As unidades de contexto são totalmente conectadas com todos os neurônios da camada escondida.

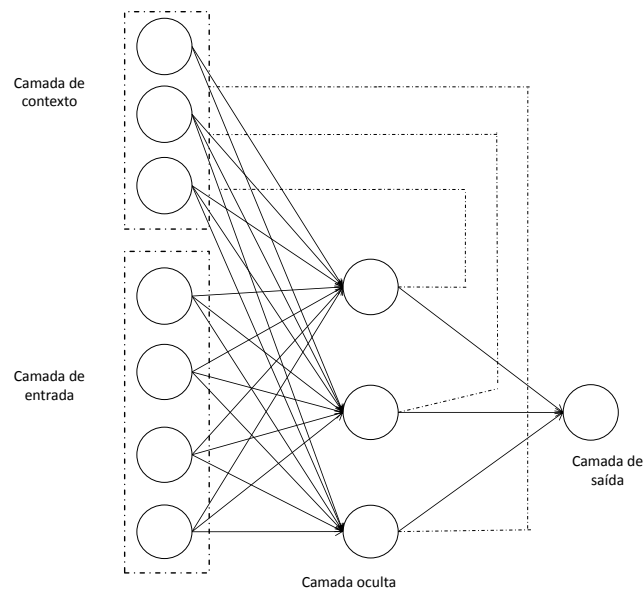


Figura 2.4 - Arquitetura da rede recorrente Elman.

2.8 Processo de Aprendizagem

A principal propriedade de uma rede neural está na habilidade de aprender através de exemplos, fazer interpolações e extrapolações do aprendizado adquirido e, especialmente, melhorar o seu desempenho através da aprendizagem. A capacidade de adquirir conhecimento é uma das mais importantes qualidades destas estruturas. Este é um processo iterativo de ajustes aplicados a seus pesos sinápticos e níveis de bias (HAYKIN, 1998).

No contexto de redes neurais, o processo de aprendizagem, pode ser definido como um conjunto de regras bem definidas para determinar a intensidade das conexões

entre os neurônios, produzindo a solução de um determinado problema de aprendizado. Diversos métodos para treinamento de redes neurais foram desenvolvidos, podendo estes serem agrupados em dois principais paradigmas de aprendizagem: aprendizado supervisionado e aprendizado não supervisionado (HAYKIN, 1998).

- **Aprendizado supervisionado:** este é o método de aprendizado mais comum no treinamento das redes neurais. A resposta desejada é fornecida por um supervisor externo. O objetivo é ajustar os parâmetros da rede, de forma a encontrar uma ligação entre os pares de entrada e saída fornecidos. A RNA tem sua saída calculada comparada com a resposta desejada, recebendo informações do supervisor sobre o erro da resposta atual. Os pesos das conexões são ajustados para minimizar o erro. O exemplo mais popular de algoritmos de aprendizado supervisionado é a regra delta ou correção de erros. Para a rede de múltiplas camadas, o algoritmo de aprendizagem utilizado é o retropropagação de erros ou (*backpropagation*) (WIDROW, 1962).
- **Aprendizado não supervisionado:** este aprendizado não há um supervisor (resposta desejada) para indicar a saída desejada aos padrões de entrada. Para este tipo de aprendizado, apenas os padrões de entrada estão disponíveis para rede, ao contrário do aprendizado supervisionado. Algoritmos de aprendizagem como o *Hebbiano* e *Competitivo* são exemplos de algoritmos não supervisionados e são especificados em Fukushima (1975), Grossberg (1998), Hebb (1949) e, Kohonen (1982).

Neste trabalho, foi utilizado o aprendizado supervisionado, apresentando o algoritmo de treinamento retropropagação do erro (*backpropagation*), o qual é a alternativa mais utilizada para treinamento de redes neurais do tipo perceptron de múltiplas camadas e rede recorrente Elman.

2.9 Treinamento

Usualmente, o conjunto de dados é dividido em três subconjuntos: o primeiro nomeado de treinamento, é composto por exemplos que serão utilizados na fase de aprendizagem (treinamento) da rede neural; o segundo subconjunto chamado de validação, é composto por exemplos que serão utilizados na tarefa de verificar a capacidade de generalização da rede e, finalmente, o conjunto de teste utilizado para testar o comportamento da rede neural com dados desconhecidos, ou seja, dados que nunca foram vistos antes pela rede neural.

A determinação dos pesos de conexão está ligado à identificação da topologia da rede neural. O algoritmo usado para ajustar os pesos de conexões em uma rede surgiu pela primeira vez em um processo de aprendizagem desenvolvido por Rosenblatt (1958) para o modelo perceptron.

O treinamento consiste na apresentação dos exemplos do conjunto de treinamento ao sistema. A quantidade de neurônios na camada de entrada é concedida pelo problema a ser tratado. No entanto, a quantidade de neurônios na camada intermediária é característica do projeto. Quando a rede neural possui um alto número de neurônios na camada intermediária, aumenta-se a capacidade de mapeamento não-linear da rede. Entretanto, quanto maior esse número, o modelo pode sobre ajustar aos dados, ocasionando um *overfitting*, dificultando sua capacidade de generalização. Por outro lado, uma rede com poucos neurônios na camada intermediária, pode não ser capaz de realizar o mapeamento desejado, o que é denominado de *underfitting* (HAYKIN, 1998).

2.9.1 Treinamento: Algoritmo de retropropagação do erro

O algoritmo de retropropagação (em inglês, *backpropagation*), é um algoritmo de aprendizado supervisionado, baseado no processo de aprendizado por correção de erros, o qual busca minimizar a diferença entre a soma ponderada das entradas com a saída calculada pela rede neural e a saída desejada, ou seja, o erro da resposta atual da rede. Sua função é encontrar na superfície de erro um mínimo global. Esse método de aprendizado por correção de erros também pode ser chamado de regra delta.

O treinamento da rede utilizando algoritmo de retropropagação de erro ocorre em duas fases (HAYKIN, 1998):

- Propagação (*forward*): consiste na ativação da rede, os pesos sinápticos são fixos, e a resposta da rede é calculada. Os sinais na camada de entrada são propagados camada por camada, até que o vetor de saída seja obtido na última camada. A fase para frente do algoritmo é análogo ao processo de filtragem do algoritmo de mínimos quadrados.
- Retropropagação (*backward*): os ajustes dos pesos sinápticos são calculados com o intuito de minimizar a soma dos erros quadráticos Equação 2.7, descrita a seguir. O sinal de erro é retropropagado da camada de saída para cada elemento da camada intermediária anterior que contribui diretamente

para a formação da saída. Este processo se repete, camada por camada, até que cada elemento da rede receba um sinal de erro que descreva sua contribuição relativa para o erro total. A saída produzida pela RNA é subtraída da resposta desejada através do aprendizado por correção de erros, para produzir um sinal de erro. A partir desse sinal, os pesos sinápticos das conexões são ajustados para cada elemento de modo a fazer a rede convergir.

As principais notações usadas no algoritmo de retropropagação são apresentadas a seguir (HAYKIN, 1998):

- Os índices i , j , e k , se referem a diferentes neurônios na rede; com os sinais se propagando através da rede da esquerda para a direita, o neurônio j se encontra em uma camada à direita do neurônio i , e o neurônio k se encontra em uma camada à direita do neurônio j , quando o neurônio j é uma camada intermediária;
- O símbolo $\varepsilon(n)$ representa a soma dos erros quadráticos ou a energia do erro na iteração n ;
- O símbolo $e_j(n)$ se refere ao sinal de erro na saída do neurônio j , para a iteração n ;
- O símbolo $d_j(n)$ representa a saída desejada para o neurônio j , sendo usada para calcular $e_j(n)$;
- O símbolo $y_j(n)$ se refere a saída do neurônio j , na iteração n ;
- O símbolo $w_{ji}(n)$, representa o peso sináptico conectando a saída do neurônio i à entrada do neurônio j , na iteração n . A correção aplicada a este peso na iteração n é representada por $\Delta w_{ji}(n)$;
- O símbolo $v_j(n)$, constitui o sinal aplicado à função de ativação associado com o neurônio j ;
- O símbolo $\varphi(\cdot)$, representa a função de ativação, que descreve a relação funcional de entrada-saída da não linearidade associada ao neurônio j ;
- O símbolo η , se refere ao parâmetro da taxa de aprendizagem;
- O símbolo δ , consiste no gradiente local.

O sinal de erro na saída do neurônio j , na iteração n é definido por:

$$e_j(n) = d_j(n) - y_j(n), \quad (2.5)$$

o neurônio j é um nó de saída.

O valor instantâneo da energia total do erro para o neurônio j é obtido somando os termos $\frac{1}{2}e_j^2(n)$ de todos os neurônios da camada de saída, os quais são os únicos neurônios visíveis para os quais os sinais de erro podem ser calculados diretamente, como descrito em:

$$\varepsilon(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n), \quad (2.6)$$

em que C é um conjunto que inclui todos os neurônios da camada de saída da rede. Considerando que N representa o número total de padrões contidos no conjunto de treinamento. A energia média do erro quadrático é obtida somando-se os ε_n para todos os n e então normalizando em relação ao tamanho do conjunto N , como apresentado por:

$$\varepsilon_{med} = \frac{1}{N} \sum_{n=1}^N \varepsilon(n). \quad (2.7)$$

A energia instantânea do erro ε_n , e conseqüentemente a energia média do erro quadrático ε_{med} é uma função de todos os parâmetros livres (pesos sinápticos e níveis de bias) da rede. Para um determinado conjunto de treinamento, ε_{med} representa a função de custo como uma medida do desempenho de aprendizagem. O objetivo do processo de aprendizagem é ajustar os pesos sinápticos e níveis de bias da rede para minimizar o erro ε_n .

O campo local induzido $v_j(n)$ produzido na entrada da função de ativação associada ao neurônio j é portanto:

$$v_j(n) = \sum_{i=0}^m w_{ji}(n)y_i(n), \quad (2.8)$$

em que m é o número total de entradas (excluindo o bias) aplicadas ao neurônio j . O peso sináptico w_{j0} é igual ao bias b_j aplicado ao neurônio j . Desta forma, o sinal $y_j(n)$ que aparece na saída do neurônio j na iteração n é:

$$y_j(n) = \varphi_j(v_j(n)). \quad (2.9)$$

O algoritmo de retropropagação aplica uma correção $\Delta w_{ji}(n)$ ao peso sináptico

$w_{ji}(n)$ usando a regra delta pela equação:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial w_{ji}(n)}, \quad (2.10)$$

em que η é o parâmetro da taxa de aprendizagem do algoritmo de retropropagação do erro.

O uso do sinal negativo na [Equação 2.10](#) indica a descida do gradiente no espaço de pesos, ou seja, busca uma direção para a mudança de peso que reduza o valor de $\varepsilon(n)$. Quanto menor o valor de η de uma interação para a outra, menor será a variação nos pesos sinápticos, porém o processo de aprendizagem é mais lento. Por outro lado, quanto maior o valor de η , a aprendizagem é mais rápida, e as modificações nos pesos podem tornar a rede instável. Uma forma de aumentar a taxa de aprendizado evitando o problema da instabilidade é modificar a regra delta incluindo o termo *momentum* α ([HAYKIN, 1998](#)).

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n), \quad (2.11)$$

em que α é o termo momento. Para que o ajuste $\Delta w_{ji}(n)$ seja convergente, a constante *momentum* deve ser $0 \leq \alpha < 1$.

O gradiente local $\delta_j(n)$ é definido por:

$$\begin{aligned} \delta_j(n) &= -\frac{\partial \varepsilon(n)}{\partial v_j(n)} \\ &= \frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \\ &= -e_j(n) \varphi'_j(v_j(n)). \end{aligned} \quad (2.12)$$

O gradiente local aponta para as modificações necessárias nos pesos sinápticos. A partir da [Equação 2.12](#), o gradiente local $\delta_j(n)$ para o neurônio de saída j é igual ao produto do sinal de erro $e_j(n)$ correspondente para aquele neurônio pela derivada $\varphi'_j(v_j(n))$ da função de ativação associada ([HAYKIN, 1998](#)).

A partir das [Equações 2.11](#) e [2.12](#) um fator chave envolvido no cálculo do ajuste de peso $\Delta w_{ji}(n)$ é o sinal de erro $e_j(n)$ na saída do neurônio j . Desta forma, é possível identificar dois casos distintos ([HAYKIN, 1998](#)).

Primeiro caso: o neurônio j é um nó de saída

Quando o neurônio j é um nó localizado na camada de saída da rede, este é suprido com uma resposta desejada. Assim, utiliza-se a [Equação 2.5](#) para calcular o sinal de erro $e_j(n)$ associado com este neurônio. Calculado o $e_j(n)$, determina-se diretamente o gradiente local $\delta_j(n)$, usando a [Equação 2.12](#).

Segundo caso: o neurônio j é um nó oculto

Quando o neurônio j é um nó oculto, não existe uma resposta especificada para aquele neurônio. Portanto, o sinal de erro para um neurônio oculto deve ser determinado recursivamente, em termos dos sinais de erro de todos os neurônios aos quais o neurônio oculto está conectado. A partir deste ponto é onde o desenvolvimento do algoritmo de retropropagação se torna complicado ([HAYKIN, 1998](#)). De acordo com a [Equação 2.12](#), é possível redefinir o gradiente local $\delta_j(n)$ para o neurônio oculto j como:

$$\begin{aligned}\delta_j(n) &= -\frac{\partial \epsilon(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \\ &= \frac{\partial \epsilon(n)}{\partial y_j(n)} \varphi'_j(v_j(n)),\end{aligned}\tag{2.13}$$

em que o neurônio j é oculto.

A seguir, a regra da cadeia é utilizada para a derivada parcial $\partial \epsilon(n)/\partial y_j(n)$

$$\frac{\partial \epsilon(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)}.\tag{2.14}$$

Na [Equação 2.15](#), é a derivada parcial desejada:

$$\begin{aligned}\frac{\partial \epsilon(n)}{\partial y_j(n)} &= -\sum_k e_k(n) \varphi'_k(v_k(n)) w_{kj}(n) \\ &= -\sum_k \delta_k(n) w_{kj}(n),\end{aligned}\tag{2.15}$$

em que na segunda linha, é utilizada a definição do gradiente local $\delta_k(n)$ dada na [Equação 2.8](#), com o índice k substituído por j .

Finalmente, utilizando as [Equações 2.15](#) e [2.13](#), tem-se a fórmula de retropropagação para o gradiente local $\delta_j(n)$ como descrito em ([HAYKIN, 1998](#)):

$$\delta_j(n) = \varphi'_k(v_k(n)) \sum_k \delta_k(n) w_{kj}(n).\tag{2.16}$$

2.9.2 Treinamento: meta-heurística MPCA

O ajuste dos pesos sinápticos é normalmente realizado através do algoritmo *backpropagation* (ou regra delta generalizada) (WILLIAMS; HINTON, 1986), este é um método que utiliza o gradiente descendente do erro de classificação da rede como informação de retorno (*feedforward*) para a adaptação dos pesos. Originalmente, o algoritmo *backpropagation* provou sua utilidade no treinamento de redes supervisionadas na resolução de problemas de classificação e mapeamento estático de funções não-lineares. Entretanto, possui convergência muito lenta e pode facilmente ficar preso em mínimos locais, tais como outros métodos locais baseados no gradiente do erro da rede neural (LEVENBERG, 1944; MARQUARDT, 1963).

O processo de treinamento supervisionado de redes perceptron de múltiplas camadas é equivalente a um problema de otimização não-linear, em que, consiste em encontrar os pesos de conexão minimizando uma função de custo. O ajuste dos pesos sinápticos pode ser formulado como um problema de otimização, em que cada solução mantém um conjunto de valores para os pesos sinápticos das conexões, representado em cada ponto no espaço de busca.

Neste contexto, foram empregados muitos trabalhos de pesquisa que utilizaram meta-heurísticas empregadas à otimização de redes neurais na tentativa de obter parâmetros adequados para os pesos das conexões evitando ficar preso nos mínimos locais. Muitos algoritmos de meta-heurísticas têm atraído grande atenção da comunidade científica. Usualmente, os algoritmos dessa classe são inspirados em algum fenômeno da natureza. Dentre os diversos algoritmos existentes para o problema de otimização dos pesos das conexões das redes neurais, destacam-se o Algoritmo Genético (YAO, 1999), o recozimento simulado (SA *Simulated Annealing*) (KIRKPATRICK, 1984), Busca Tabu (GLOVER, 1986), Otimização por Colônia de Formigas (DORIGO et al., 1996).

Neste trabalho, a meta-heurística MPCA realiza o papel do algoritmo de treinamento *backpropagation* efetuando o ajuste dos principais parâmetros livres da rede e os pesos sinápticos do treinamento (ver Capítulo 5). Nesta abordagem, os parâmetros e pesos são definidos por um processo de otimização, em que o desempenho da rede neural corresponde à minimização de uma função objetivo.

3 ALGORITMOS DE OTIMIZAÇÃO

Este capítulo mostra os conceitos básicos usados em otimização mono-objetivo, multiobjetivo e meta-heurística. Em seguida, são apresentados os algoritmos *Non-Dominated Sorting Genetic Algorithm* (NSGA-II) – um algoritmo genético multiobjetivo – e o Algoritmo de Colisão de Múltiplas Partículas, os quais foram selecionados para serem utilizados como otimizadores na metodologia proposta.

3.1 Otimização

A otimização é um paradigma fundamental em muitas aplicações, tais como engenharia, atividades empresariais, na indústria, entre outras. Naturalmente, os objetivos de otimização em todas as aplicações citadas acima, são a minimização do consumo de energia e custos, ou a maximização do lucro, da produção, do desempenho e a eficiência. Na realidade, os recursos (tempo e dinheiro) são sempre limitados e, conseqüentemente, a otimização deste é um fator essencial (SINGIRESU, 2009).

Considerando o espaço de busca a ser explorado, os problemas de otimização são compostos por diversos campos, porém há algumas intersecções entre seus subcampos. Algumas sub-áreas são (BECCENERI, 2013):

- Programação linear: a função custo (objetivo) é linear e há um conjunto de restrições relacionadas às variáveis da função, que são especificadas como igualdades ou desigualdades lineares;
- Otimização contínua: as variáveis assumem valores reais ou contínuos;
- Otimização combinatória (discreta): as variáveis assumem valores discretos ou inteiros;
- Otimização mista: com variáveis inteiras e contínuas ao mesmo tempo;
- Otimização estocástica: estuda os casos em que as restrições ou os parâmetros dependem de variáveis aleatórias.

Goh e Tan (2009) apresentam alguns conceitos básicos de otimização:

- Variáveis de decisão: formam um conjunto de n parâmetros cujos valores representam uma solução para um problema de otimização. Elas podem ser contínuas (reais), inteiras ou discretas (valores compreendidos dentro de um certo conjunto fixo).

- Restrições: são funções das variáveis de decisão que delimitam o problema e validam as soluções. Pode-se dizer que as restrições esboçam o contorno da região onde se encontra o conjunto de soluções factíveis do problema. As restrições dividem-se em dois grupos: restrições laterais onde são efetuadas diretamente sobre as variáveis de decisão, limitando seus valores; e a restrição de comportamento que consiste nas condições desejáveis de limites de tensões, deslocamentos, frequências naturais de vibração, entre outros.
- Funções objetivo: são critérios de avaliação para determinar a qualidade de uma solução, ou seja, é a função que se deseja otimizar. Na otimização multiobjetivo existem duas ou mais funções objetivo em cada problema.
- Ponto Ótimo: é o ponto caracterizado pelo vetor $x = (x_1, x_2, \dots, x_n)$ formado pelas variáveis de decisão que otimizam as funções objetivo e satisfazem as restrições.
- Valor Ótimo: é o valor da função objetivo $f(x)$ no ponto ótimo.
- Solução Ótima: a solução ótima para cada função j é o par formado pelo ponto ótimo e o valor ótimo $[x^*, f(x^*)]$.

3.1.1 Otimização mono-objetivo

Em um problema de otimização mono-objetivo, uma única medida de desempenho é considerada entre as várias possíveis, que resultará em um conjunto exclusivo de parâmetros que fornece uma solução ótima, correspondente às soluções extremas (mínimas ou máximas) da função objetivo do problema em questão.

Usualmente, um problema de otimização pode ser escrito como:

(i) Minimizar (ou maximizar)

$$f(x) \tag{3.1}$$

(ii) sujeito às restrições não-lineares:

$$g_i(x) \leq 0, i = 1, 2, \dots, m \tag{3.2}$$

$$h_j(x) = 0, j = 1, 2, \dots, p \tag{3.3}$$

$$x = (x_1, x_2, \dots, x_n)^T \quad (3.4)$$

Na expressão 3.1, $f(x)$ é a função que se deseja otimizar. Esta função é chamada de função objetivo do problema e pode ser minimizada ou maximizada.

As expressões 3.2 e 3.3 representam as restrições do problema. E a expressão 3.4 é o vetor com n variáveis de decisão. A variável x representa uma solução do conjunto e que para ser válida deve respeitar as restrições $g(x)$ e $h(x)$ do problema.

3.1.2 Otimização multiobjetivo

Um problema de otimização multiobjetivo, geralmente, consiste em minimizar (ou maximizar), de forma simultânea, um conjunto de critérios (objetivos) satisfazendo um conjunto de restrições, que resultará em vários conjuntos de parâmetros que fornecem um conjunto de soluções ótimas (conjunto de Pareto), do qual, espera-se, obter-se uma resposta desejada.

Muitos problemas do mundo real envolvem a otimização simultânea de vários objetivos conflitantes e restrições que são difíceis, se não impossível, para resolver sem a ajuda de algoritmos de otimização. O que torna a otimização multiobjetivo tão desafiadora é que, na presença de especificações conflitantes, há uma solução ideal para todos os objetivos (GOH; TAN, 2009).

Neste contexto, otimização é uma realidade que é aplicada em quase todas as áreas e podem apresentar mais de um objetivo, em que na maioria das vezes, são conflitantes entre si, ou seja, a melhora de um ou mais objetivos causam, conseqüentemente, a deterioração de outros.

3.1.2.1 Formalização do problema de otimização multiobjetivo

A formulação geral do problema de otimização multiobjetivo pode ser expressa (Deb, 2004) da seguinte maneira:

(i) Otimizar:

$$F(x) = (f_1(x), f_2(x), \dots, f_k(x)), \quad k = 1, \dots, l ; \quad l \geq 2 \quad (3.5)$$

(i) Sujeito a:

$$g_j(x) \leq 0, \quad j = 1, \dots, q \quad (3.6)$$

$$h_j(x) = 0, \quad j = q + 1, \dots, m \quad (3.7)$$

$$v_i \leq x_i \leq u_i, \quad i = 1, \dots, n \quad (3.8)$$

A solução $x = (x_1, x_2, \dots, x_n)^T$ é um vetor com x variáveis de decisão, que satisfaça às restrições de desigualdade (3.6), restrições de igualdades (3.7), e restrições laterais (3.8) e que otimize simultaneamente as funções objetivo 3.5.

As restrições acima estabelecem um espaço de soluções factíveis das variáveis de decisão e as que não satisfizerem ao menos uma das restrições, são chamadas de não-factíveis, que é denominado conjunto eficiente ou Pareto ótimo. Todos os valores das funções objetivo, que fazem parte deste conjunto ótimo, são soluções atrativas para o problema multiobjetivo sob análise.

3.1.2.2 Conjunto e fronteira de Pareto

Em um problema de otimização multiobjetivo não existe uma única solução ótima, sendo, neste caso, o resultado deste processo um conjunto de soluções. É comum existirem várias soluções, tais que, ao mudar de uma solução para outra, percebe-se a melhora obtida em uma função objetivo, e compete ao tomador de decisões adotar a mais adequada para solução do problema.

Conjunto Pareto-ótimo é formado pelo conjunto de todas as soluções não-dominadas, dentre as soluções factíveis. Enquanto que a fronteira de Pareto é formada pelos pontos no espaço das funções objetivo que corresponde ao conjunto Pareto-ótimo.

De um modo geral, uma solução x_1 é dita pertencer ao conjunto de Pareto, se não existir nenhuma outra solução x_2 , capaz de melhorar um dos objetivos do problema (em relação a x_1) sem, simultaneamente, piorar pelo menos um dos demais. Todas as soluções pertencentes ao conjunto de Pareto são ditas não-dominadas. Segundo Deb (2001), o conceito de dominância pode ser definido como:

Definição 1 - Uma solução x_1 domina outra solução x_2 quando as duas condições seguintes são satisfeitas:

- A solução x_1 é pelo menos igual a x_2 em todas as funções objetivos;
- A solução x_1 é estritamente melhor do que x_2 em pelo menos um objetivo.

Essas propriedades caracterizam a relação de dominância como uma relação de ordem parcial estrita (DEB, 2001).

Definição 2 - Um conjunto de soluções x_1 domina outro conjunto de soluções x_2 para todo $k \in \{1, 2, \dots, l\}$, tal que $f_k(x_1) \leq f_k(x_2)$, existe $j \in \{1, 2, \dots, l\}$ tal que $f_j(x_1) < f_j(x_2)$.

Quando uma solução x_1 domina outra solução x_2 , o mais comum é dizer que x_1 é não dominada por x_2 .

A Figura 3.1 ilustra um problema de otimização com dois objetivos e com cinco soluções diferentes mostrados no espaço objetivo. Neste exemplo, as duas funções f_1 e f_2 precisam ser minimizadas. Em geral, encontrar a melhor solução em relação à ambos objetivos é uma tarefa complexa, uma vez que ambas funções objetivo são relevantes. Assim, é aplicado o conceito de dominância na decisão da melhor solução.

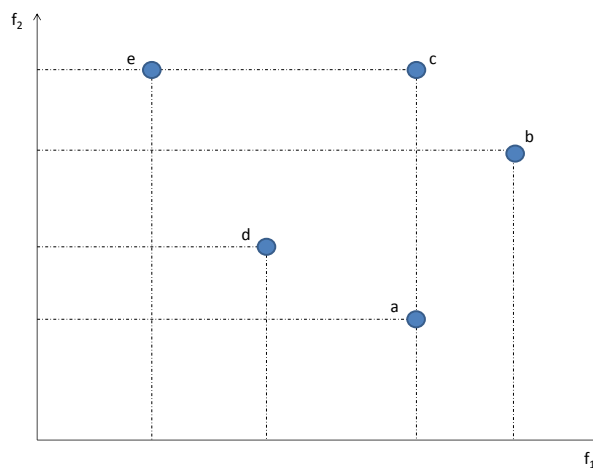


Figura 3.1 - Relação de dominância em uma minimização entre duas soluções.

Fonte: Adaptado de (DEB, 2001).

Pelas definições de dominância, verifica-se que a solução a domina as soluções b e c , as soluções a e d são não-dominadas entre si e o conjunto $\{a, d, e\}$ é um conjunto de soluções não-dominadas que dominam as soluções b e c . Assim, pelo menos um elemento do conjunto $\{a, d, e\}$ domina os elementos de b e c (DEB, 2001).

3.2 Algoritmos Meta-heurísticos

A meta-heurística foi introduzida por Glover (1986) ao conceitualizar a existência de uma heurística imposta a outra heurística. O termo “heurística” é a arte de encontrar ou inventar, enquanto o prefixo “meta” significa além de, em um nível superior, ou seja, expressa o conceito de nível superior ou maior generalidade. Antes deste termo ser amplamente adotado, meta-heurísticas eram chamadas de “heurísticas moderna” (REEVES, 1993).

Algoritmos meta-heurísticos são muitas vezes inspirados na natureza, e estão entre os algoritmos mais utilizados em problemas de otimização. A eficiência desses algoritmos pode ser atribuída ao fato de reproduzirem as melhores características da natureza, especialmente, a seleção dos mais aptos em sistemas biológicos que têm evoluído pela seleção natural. As meta-heurísticas compõem uma área de pesquisa bastante ativa, uma vez que são aplicáveis às mais diversas áreas, e nem sempre pertencentes a ciências exatas.

Blum e Roli (2003) apresentam algumas propriedades que caracterizam as meta-heurísticas:

- Meta-heurísticas são estratégias que “guiam” o processo de busca (pesquisa);
- O objetivo é explorar de forma eficiente o espaço de busca, a fim de encontrar soluções (quase) ótimas;
- Técnicas que constituem algoritmos meta-heurísticos variam de simples procedimentos de busca local à complexos processos de aprendizagem;
- Os algoritmos meta-heurísticos são aproximados e geralmente não-determinísticos;
- Podem incorporar mecanismos para evitar ficar presos em áreas fechadas do espaço de buscas;
- Meta-heurísticas não são específicas para certos problemas;
- Meta-heurísticas podem fazer o uso do conhecimento específico do domínio na forma de heurísticas que são controladas pela estratégia de nível superior.

Os conceitos de diversificação (*diversification*) e intensificação (*intensification*) são de grande relevância para meta-heurística. O conceito de diversificação se refere à exploração do espaço de busca, enquanto que o conceito de intensificação refere-se a exploração da experiência de pesquisa acumulada (BLUM; ROLI, 2003).

Existem diferentes maneiras de classificar e descrever algoritmos meta-heurísticos, dependendo de suas características, tais como Blum e Roli (2003), Yang (2010), Yang e Koziel (2011) descrevem:

- Inspirada na natureza ou não-inspirada natureza: a maneira mais intuitiva de classificar baseia-se na origem do algoritmo. Há algoritmos inspirados na natureza, tais como, Algoritmos Genéticos (*Genetic Algorithm*) (GOLDBERG, 1989), Otimização por Colônia de Formigas (*Ant Colony Optimization*) (DORIGO et al., 2006), Algoritmos de Abelhas (*Bee Algorithms*), Sistema Imunológico Artificial (*Artificial Immune System*) (FARMER et al., 1986); e os não inspirados na natureza, como a Busca Tabu (*Tabu Search*, (TS)) (GLOVER, 1990) e Busca local Iterada (*Iterated Local Search (ILS)*).
- Baseada em população ou busca de ponto único: algoritmos baseados em busca de ponto único, trabalham com soluções individuais chamados de métodos de trajetória e englobam meta-heurísticas baseadas em busca local, estes incluem a TS, ILS e Busca em Vizinhança Variável (*Variable Neighborhood Search (VNS)*). Por outro lado, algoritmos baseados em população executam os processos de busca que descrevem a evolução de um conjunto de pontos no espaço de busca.
- Função objetivo dinâmica ou estática: meta-heurísticas também podem ser classificadas de acordo com o uso da função objetivo. Enquanto que alguns algoritmos preservam a função objetivo na representação problema, outros, como a Busca Local Guiada (*Guided Local Search*), modificam a função objetivo durante o processo de busca. A ideia dessa abordagem é escapar de mínimos locais, modificando a paisagem de busca (em inglês *search landscape*). Assim, durante o processo de busca a função objetivo é alterada com intuito de tentar incorporar informações coletadas durante o processo de busca.
- Estrutura de vizinhança única ou variável: a maioria dos algoritmos meta-heurísticos trabalham em uma única estrutura de vizinhança. Em outras palavras, a topologia da paisagem de busca não altera no decorrer do algo-

ritmo. Outras meta-heurísticas, tais como o VNS, utiliza um conjunto de estruturas de vizinhança, em que permite a possibilidade de diversificar a busca em diferentes formas de aptidão.

- Métodos com uso de memória ou sem memória: uma característica muito importante para classificar meta-heurísticas é a utilização de memória adaptativa. Um algoritmo sem memória executa um processo em que as informações que são utilizadas exclusivamente para determinar a próxima ação é o estado atual do processo de busca. Existem diversas maneiras de utilizar o uso de memória. Normalmente, utiliza-se uma diferenciação entre o uso de curto prazo e memória de longo prazo. O primeiro mantém o controle de movimentos realizados recentemente, soluções visitadas ou as decisões tomadas. O segundo é um acúmulo de parâmetros de síntese sobre a pesquisa.

Nas duas seções seguintes, uma descrição detalhada das meta-heurísticas Algoritmo de Colisão de Múltiplas Partículas (*Multiple Particle Collision Algorithm* (MPCA)) e o Algoritmo Genético Multiobjetivo (*Non-dominated Sorting Genetic Algorithm* (NSGA-II)), são apresentadas.

3.3 Algoritmo de Colisão de Partículas (PCA)

O Algoritmo de Colisão de Partículas (PCA) proposto por (SACCO; OLIVEIRA, 2005), foi inspiração no Algoritmo de Recozimento Simulado (*Simulated Annealing* (SA)) (METROPOLIS et al., 1953), que define um algoritmo de rejeição usado para gerar uma sequência de amostras de uma distribuição de probabilidade de amostragem direta.

A estrutura do PCA se assemelha ao SA, primeiramente uma solução inicial é determinada e em seguida esta solução é modificada através de uma perturbação estocástica. As qualidades dessas possíveis soluções são comparadas e então é decidido pela manutenção ou alteração da solução atual por outra solução potencial.

O algoritmo PCA é inspirado na física das reações de colisão de partículas em um reator nuclear, em particular nos comportamentos de espalhamento (*scattering*) e absorção (*absorption*). O uso do algoritmo PCA, tem sido utilizado com sucesso em diversos estudos de caso de otimização, desde problemas de testes (*benchmark*) quanto em problemas de aplicações reais (SACCO et al., 2006; SACCO et al., 2008; SACCO; OLIVEIRA, 2005). Em Knupp et al. (2009) foi implementada uma versão

baseada em uma análise de sensibilidade prévia, na qual apenas os dados com maior sensibilidade são utilizados.

A Figura 3.2 mostra o principal trecho do algoritmo PCA para maximização de uma função.

```
Gera uma solução inicial: Old-Config
Best-Fitness = Fitness(Old-Config)
Para n = 0 até # of iterações
    Perturbation(.)
    Se Fitness(New-Config) > Fitness(Old-Config)
        Se Fitness(New-Config) > Best-Fitness
            Best-Fitness = Fitness(New-Config)
        Fim-se
        Old-Config = New-Config
        Exploration(.)
    Senão
        Scattering(.)
    Fim-se
Fim-Para
```

Figura 3.2 - Algoritmo PCA.

Fonte: (LUZ, 2012).

A estrutura do algoritmo PCA inicia-se com uma seleção de uma solução inicial (**Old-Config**), que sofre uma modificação por uma perturbação estocástica através da função **Perturbation()**, descrita na Figura 3.3, e conduz para a construção de uma nova solução (**New-Config**). As qualidades das duas soluções são calculadas através da função **Fitness()** e comparadas, a partir da nova solução pode ser aceita, ou não, dependendo de sua performance frente a solução anterior. Se a nova solução for melhor ela é aceita e um procedimento de exploração local é ativado pela função (**Exploration**), caso contrário, o procedimento de espalhamento (**Scattering**) é iniciado, tal como demonstrado na Figura 3.5 (LUZ, 2012).

A função perturbação (**Perturbation**) executa uma variação aleatória da solução dentro de um intervalo definido, permitindo a visita em diferentes regiões no espaço de busca. Para garantir que as perturbações impostas ao algoritmo não levem à construção de uma solução na qual um de seus pontos esteja fora do espaço válido de buscas, uma verificação destes limites é realizada constantemente.

```

Perturbation(.)
  Para i = 0 até (Dimensões-1)
    Superior = Limite-superior[i]
    Inferior = Limite-inferior[i]
    Aleatório = Random(0,1)
    New-Config[i] = Old-Config[i] + ((Superior - Old-Config[i] *
    Aleatório) - ((Old-Config[i] - Inferior)*(1-Aleatório))
    Se New-Config[i] > Superior
      New-Config[i] = Limite-superior[i]
    Senão
      Se New-Config[i] > Inferior
        New-Config[i] = Limite-inferior[i]
    Fim-se
  Fim-se
Fim-Para
Retorna

```

Figura 3.3 - Função para perturbação no PCA.

Fonte: (LUZ, 2012).

Na função exploração (**Exploration**), pequenas perturbações estocásticas são aplicadas à solução através da função (**Small-Perturbation**), esta solução é aceita e um esquema de exploração local é ativado, como apresentada na Figura 3.4.

```

Exploration(.)
  Para n = 0 até # de iterações
    Small-Perturbation()
    Se Fitness(New-Config) > Fitness(Old-Config)
      Old-Config = New-Config
    Fim-se
  Fim-Para
Retorna

```

Figura 3.4 - Função para exploração no PCA.

Fonte: (LUZ, 2012).

A Figura 3.5 demonstra o procedimento da função de espalhamento (**Scattering**), um esquema estocástico que tenta evitar a prisão do algoritmo em uma região de ótimo local no espaço de buscas. Este procedimento é baseado no esquema de *Me-*

```

Scattering(.)
    Pscattering = 1 - Fitness(New-Config) / Best-Fitness
    Se Pscattering > Random(0,1)
        Old-Config = Solução aleatória
    Senão
        Exploration()
    Fim-Se
Retorna

```

Figura 3.5 - Função para espalhamento no PCA.

Fonte: (LUZ, 2012).

tropolis, i.e. uma solução pode ser aceita, com certa probabilidade, mesmo sendo sua qualidade menor do que aquela da solução antiga. Tal flexibilidade pode evitar a convergência para ótimos locais (METROPOLIS et al., 1953).

3.3.1 Algoritmo de Colisão de Múltiplas Partículas (MPCA)

O MPCA que se constitui numa nova meta-heurística introduzida por Luz et al. (2008), é uma extensão do PCA canônico (SACCO; OLIVEIRA, 2005).

A estrutura proposta para o algoritmo MPCA, utiliza-se um conjunto de n partículas explorando, de maneira independente, porém colaborativa, o mesmo espaço. A introdução de n partículas leva a necessidade da implementação de um mecanismo de comunicação indireta entre as partículas.

Este mecanismo foi implementado via *Message Passing Interface* (MPI), em um ambiente computacional de alto desempenho, em que o melhor resultado (**Best-Fitness**) obtido pelas partículas é atualizado de maneira constante entre todas as partículas envolvidas no processo de busca (LUZ, 2012).

Este processo é chamado de *blackboard*, em que as partículas trabalham de maneira independente no processo de busca. Assim, no momento em que alguma partícula identifica uma nova melhor posição, esta partícula escreve no *blackboard* para que as outras partículas tomem conhecimento deste novo valor. O pseudo-código do MPCA é demonstrado na Figura 3.6.

O MPCA tem sido utilizado com sucesso em diversos problemas de otimização, tais como, diagnóstico de falhas (ECHEVARRÍA et al., 2012), identificação de perfil de temperatura atmosférica (SAMBATTI et al., 2014), previsão climática de precipitação

(ANOCHI; CAMPOS VELHO, 2014) e na solução de um problema inverso radiativo (Hernández Torres et al., 2015).

```
Gera uma solução inicial: Old-Config
Best-Fitness = Fitness(Old-Config)
Atualiza o Blackboard
Para n = 0 até # of iterações
  Para n = 0 até # of partículas
    Perturbation(.)
    Se Fitness(New-Config) > Fitness(Old-Config)
      Se Fitness(New-Config) > Best-Fitness
        Best-Fitness = Fitness(New-Config)
      Fim-Se
      Old-Config = New-Config
      Exploration(.)
    Senão
      Scattering(.)
    Fim-Se
  Atualiza o Blackboard
Fim-Para
Fim-Para
```

Figura 3.6 - Algoritmo de colisão de múltiplas partículas.

Fonte: (LUZ, 2012).

Os parâmetros que regulam o funcionamento do algoritmo MPCA são:

- número de iterações globais do algoritmo;
- número de iterações para as perturbações locais, correspondendo ao laço e à quantidade de chamadas à função (**Small-Perturbation**);
- função de probabilidade ($P_{scattering}$) usada para a ativação do espalhamento;
- limite superior (LS) da intensidade da perturbação na busca local, usada pela função (**Small-Perturbation**);
- limite inferior (LI) da intensidade da perturbação na busca local, usada pela função (**Small-Perturbation**);
- número de processadores ($N_{processadores}$) que serão usados;

- número de partículas ($N_{partículas}$) que explorarão o espaço de buscas;
- número de iterações ($N_{blackboard}$) que serão executadas antes da ativação do ciclo de sincronização com o *blackboard*.

3.4 Algoritmo Genético

Algoritmo Genético (AG) é uma meta-heurística inspirada na Teoria da Evolução de Darwin que simula o processo de seleção natural e a sobrevivência dos indivíduos mais aptos de uma população. Foi desenvolvido por Holland (1975) e popularizado por Goldberg (1989).

A evolução no AG é normalmente iniciada a partir de um conjunto de soluções geradas aleatoriamente e é realizada por meio de três principais operadores: seleção, cruzamento e mutação, ao longo das gerações. A cada geração, a adaptabilidade de cada solução na população é avaliada, alguns indivíduos são selecionados para a próxima geração, e recombinados (mutação) para formar uma nova população. A nova população é, então, utilizada como entrada para a próxima geração do algoritmo.

3.4.1 Operadores genéticos em AG

- Indivíduo: é uma solução candidata cujo código genético é, no algoritmo genético simples, representado por uma cadeia de bits. Os indivíduos representam as possíveis soluções para o problema no espaço de busca e, é através de sua manipulação (pelo processo de evolução) que as respostas são encontradas.
- Tamanho da população: a escolha do tamanho da população é um fator importante, uma vez que, pode afetar o desempenho global e a eficiência dos AGs. Com uma população pequena o desempenho pode cair, pois o algoritmo terá pouca possibilidade de realizar cruzamentos e apenas uma parte pequena do espaço de soluções será explorada. Enquanto que, uma grande população geralmente fornece uma cobertura representativa do domínio do problema, além de prevenir convergências prematuras para soluções locais ao invés de globais.
- Método de seleção: os indivíduos são selecionados para a reprodução, formando a próxima população. A maioria dos métodos de seleção é desenvolvida para escolher preferencialmente indivíduos com maiores valores de adaptabilidade ou aptidão, embora não exclusivamente, a fim de manter a

diversidade da população. Existem vários métodos para efetuar a seleção, dentre as quais destacam-se: método da roleta, seleção por torneio, seleção por classificação (*ranking*), seleção por truncamento.

- Taxa de cruzamento (*crossover*): determina a quantidade que o cruzamento será executado. Altas taxas de cruzamento aumentam a probabilidade de convergência para um mínimo local. Por outro lado, com baixas taxas, podem tornar a convergência muito lenta.
- Taxa de mutação: determina a frequência que os cromossomos sofrerão mutação. A mutação opera sobre os indivíduos resultantes do processo de cruzamento e com uma probabilidade pré-determinada efetua algum tipo de alteração em sua estrutura. Se houver baixas probabilidades de mutação, é suficiente para prevenir que o algoritmo fique estagnado em um mínimo local. Enquanto que, com uma taxa muito alta de mutação a busca se torna essencialmente aleatória.

A mutação opera sobre os indivíduos resultantes do processo de cruzamento e com uma probabilidade pré-determinada efetua algum tipo de alteração em sua estrutura. A principal função deste operador reside no fato de que uma vez bem escolhido seu modo de atuar garante que diversas alternativas serão exploradas, mantendo assim um nível mínimo de abrangência na busca

A Figura 3.7 apresenta a estrutura básica do algoritmo genético.

O algoritmo genético, como descrito anteriormente é adequado para resolver problemas de otimização mono-objetivo. Entretanto, muitas aplicações reais envolvem a otimização de múltiplos objetivos simultaneamente.

3.4.2 Non-dominated Sorting Genetic Algorithm (NSGA-II)

O algoritmo *Elitist Non-Dominated Sorting Genetic Algorithm* (NSGA - II), desenvolvido por Deb et al. (2002), é um algoritmo evolutivo multiobjetivo, proposto como uma versão melhorada do NSGA (*Non-dominated Sorted Genetic Algorithm*) (SRINIVAS; DEB, 1994).

A principal diferença do NSGA-II e um AG simples é a forma como o operador de seleção é aplicado, sendo este operador subdividido em dois processos: ordenação pelo critério rápido por não-dominância (*fast non-dominated sorting*) e a ordenação pelo

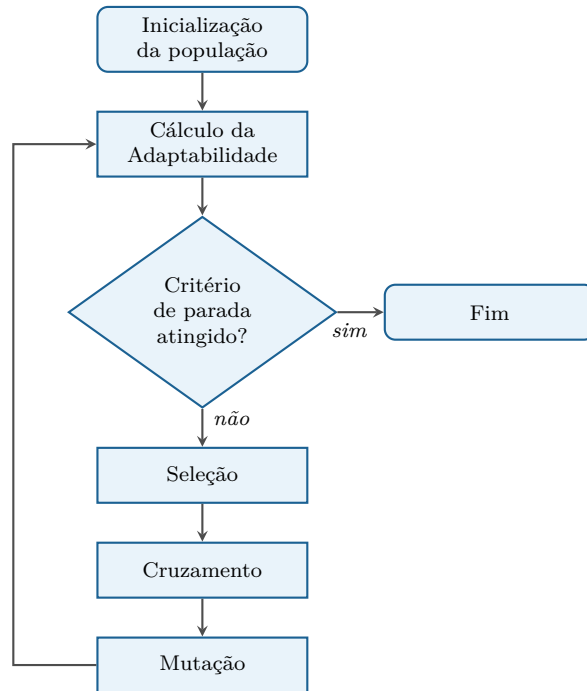


Figura 3.7 - Estrutura b sica do algoritmo gen tico.

crit rio de Dist ncia de Agrupamento (*crowding distance*). Os demais operadores s o aplicados de maneira tradicional (DEB et al., 2002).

O ordenamento de Pareto no NSGA-II   feito de uma maneira mais r pida (*fast non-dominated sorting*) em duas partes.

Primeiramente, todos os indiv duos da populaç o P s o comparados entre si a partir dos valores das funç es objetivo, para determina o do grau de domin ncia e classifica o. O grau de domin ncia de um indiv duo p da populaç o, denotado por np , corresponde ao n mero de indiv duos que dominam p . Ao t rmino da primeira parte do processo, os indiv duos que possuem o grau de domin ncia $np = 0$, ou seja, n o dominados, s o inseridos na primeira frente de Pareto (melhores indiv duos de toda a populaç o).

Na segunda parte do processo, todos os indiv duos s o separados em diferentes frentes de domin ncia cujo grau de domin ncia   diferente de zero. Cada indiv duo p em uma frente   retirado da populaç o, os graus de domin ncia dos indiv duos que s o dominados por p s o recalculados. O processo   repetido at  que n o sobrem mais indiv duos na populaç o (DEB et al., 2002).

Ao t rmino do Ordenamento de Pareto, a populaç o   classificada em M subcon-

juntos, sendo que nenhuma solução domina outra. A Figura 3.8 mostra um exemplo de fronteira de Pareto ótima; os indivíduos da primeira frente são as melhores soluções para a geração atual, enquanto os indivíduos da última frente correspondem às piores. Assim, na primeira frente estão os indivíduos que não são dominados por nenhum outro indivíduo da geração.

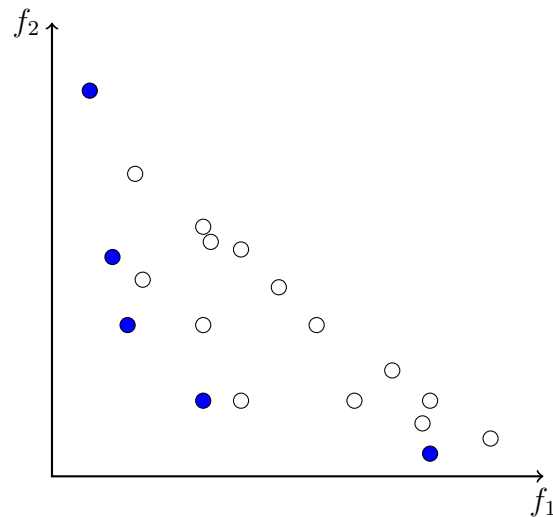


Figura 3.8 - Exemplo de uma frente de Pareto.

Para obter uma estimativa da densidade das soluções de um determinado ponto, é aplicado o segundo processo do operador de seleção do NSGA-II, denominado *crowding distance* (Distância de Aglomeração). Este é usado para garantir um espalhamento uniforme das soluções ao longo da frente de Pareto. Para melhor compreender o operador *crowding distance*, é necessário definir a métrica para estimação de densidade e o operador de comparação (DEB et al., 2002).

A estimativa da densidade de soluções em torno de uma solução da população, é obtida através do cálculo da distância média, em relação às funções objetivo, entre o ponto e seus vizinhos na mesma frente.

O operador de comparação (\prec_n) orienta o processo de seleção levando a um espalhamento uniforme em direção a uma frente de Pareto ótima. Suponha que cada indivíduo i da população possui dois atributos: classificação da fronteira (*rank* de não dominância) i_{rank} e *crowding distance* $i_{distance}$ tem-se:

$$i \prec_n j \text{ se } (i_{rank} < j_{rank}) \text{ ou } ((i_{rank} = j_{rank}) \text{ e } (i_{distance} > j_{distance})) \quad (3.9)$$

Portanto, entre duas soluções com diferentes frentes não-dominadas, aquela que possuir a solução com a menor classificação será escolhida. Enquanto que, se ambas as soluções pertencerem na mesma fronteira, aquela que estiver em uma região com o menor número de soluções vizinhas, será escolhida.

A Figura 3.9 mostra o funcionamento do NSGA-II. Inicialmente, uma população combinada $R_t = P_t + Q_t$ é formada, onde P_t representa a população atual (de pais), e Q_t representa a população de descendentes (filhos) criada a partir da aplicação dos operadores genéticos de seleção, cruzamento e mutação. Esta nova população R_t , de tamanho $2N$, em que N representa o tamanho da população será ordenada de acordo com o critério de não-dominância. A nova população P_{t+1} é formada adicionando membros da primeira fronteira até que o tamanho exceda N . Em seguida, as soluções da última fronteira aceita são sorteadas com base no operador de comparação \prec_n e os primeiros N pontos são selecionados. Essa nova população é usada para formar uma outra população Q_{t+1} , de tamanho N através do processo de seleção, cruzamento e mutação.

O Conjunto F_1 conterà as melhores soluções da população combinada. Quando o tamanho de F_1 é menor que N , todos os indivíduos deste conjunto são escolhidos para formar a nova população P_{t+1} , seguidos pelas frentes de acordo com a ordem de dominância, até completar os N indivíduos. Em caso de duas soluções estarem em uma mesma frente, a solução escolhida é aquela que apresenta o maior valor da distância de aglomeração, ou seja, a que está localizada na região menos densa de soluções (DEB et al., 2002).

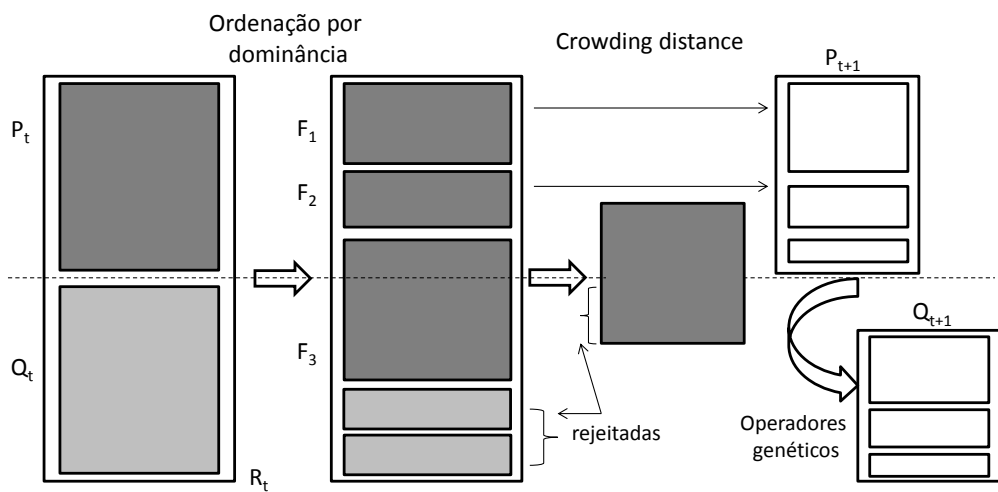


Figura 3.9 - Esquema do algoritmo NSGA-II.

Fonte: Adaptado de (DEB et al., 2002).

4 TEORIA DOS CONJUNTOS APROXIMATIVOS

Neste capítulo são apresentados e exemplificados os principais conceitos da Teoria dos Conjuntos Aproximativos, um paradigma para tratamento de informações incertas e imprecisas. Esta teoria é motivada pela necessidade prática de interpretar, representar e processar a indiscernibilidade entre elementos.

4.1 Introdução

A Teoria dos Conjuntos Aproximativos (TCA) (em inglês *Rough Set Theory*) foi proposta no início da década de 80 pelo matemático polonês Pawlak (1982), como um formalismo matemático para o tratamento de informações incertas e imprecisas.

Existem vários modelos formais disponíveis para o tratamento de incertezas contidas em base de dados, tais como a Teoria dos Conjuntos Difusos (*Fuzzy Sets Theory*) (ZADEH, 1965); a Teoria de Evidência (*Dempster-Shafer theory* (DST)) (DEMPSTER, 1967), e a Teoria das Possibilidades (*Possibility Theory*) (ZADEH, 1978).

A principal vantagem da TCA em análise de dados é a de não necessitar de alguma informação preliminar ou adicional sobre os dados, como a distribuição de probabilidade, o grau de pertinência (ZADEH, 1965), ou atribuição de probabilidade básica (DEMPSTER, 1967).

A TCA tem atraído a atenção de muitos pesquisadores, os quais contribuíram essencialmente para o seu desenvolvimento e aplicações em Inteligência Artificial, especialmente nas áreas: aprendizagem de máquina, aquisição de conhecimento, análise em tomada de decisão, descoberta de conhecimento em base de dados, sistemas especialistas, raciocínio indutivo e reconhecimento de padrões (PAWLAK, 1982).

Peters et al. (2003) propuseram o uso da TCA para classificar atributos de tempestade relacionadas à ocorrência de granizo, chuva, tornado e vento, no Canadá. Para tal, foi criada uma base de dados com 22 variáveis (atributos condicionais) e o atributo de decisão, a qual foi analisada utilizando a TCA.

Pessoa e Stephany (2014) utilizaram a TCA para reduzir os número de variáveis meteorológicas de entrada para o modelo de rede Perceptron de Múltiplas Camadas, utilizado para produzir a previsão climática de precipitação.

Aprianti e Mukhlash (2014) utilizaram duas abordagens para o tratamento de dados incompletos, a TCA e a teoria dos conjuntos difusos, utilizando dados meteorológicos

para classificar eventos de chuva ou seca. Os autores mostraram que a abordagem TCA foi mais eficiente em relação a teoria dos conjuntos difusos neste problema.

A TCA é caracterizada pelo bom formalismo matemático e simplicidade de aplicação, tornando-a um paradigma importante no tratamento de incertezas. Algumas características são levadas em consideração na abordagem TCA:

- Fornece algoritmos eficientes para encontrar padrões ocultos nos dados;
- Busca conjuntos mínimos de dados (redução de dados);
- Realiza uma avaliação no significado dos dados;
- Gera conjuntos de regras de decisão a partir de dados;
- Oferece interpretação direta dos resultados obtido.

A TCA é uma extensão da teoria dos conjuntos, que enfoca o tratamento de imperfeição da informação, por meio de uma relação de indiscernibilidade e avalia se os elementos de um conjunto são indiscerníveis, ou seja, se possuem as mesmas propriedades. Se tal relação de indiscernibilidade é verificada entre dois ou mais objetos, significa que se elegerá somente um destes objetos, reduzindo então a dimensão dos dados.

Esta relação permite o uso da TCA na construção de subconjuntos de atributos, a partir de uma base de dados, para representar o conhecimento, através de um processo de eliminação de atributos irrelevantes proporcionando a redução dos dados, através dos chamados redutos (PAWLAK, 1982).

Neste trabalho, para o processo de redução de dados, foi utilizado o sistema Rosetta (*Rough Set Toolkit for Analysis of Data* - <http://www.lcb.uu.se/tools/rosetta>). Rosetta possui um conjunto de algoritmos para análise de dados, inspirada na abordagem da TCA, construída como ferramenta de apoio aplicada em processos de mineração dados e descoberta de conhecimento em dados, abrangendo as diferentes etapas do processo (ØHRN, 1999b).

4.2 Sistemas de Informação

Na abordagem TCA, os dados são representados de forma tabular, através de um Sistema de Informação (SI). Os dados são representados no formato de tabela, em

que cada linha representa um objeto e as colunas, representam os atributos (PAL; SKOWRON, 1999).

A definição formal de um SI é um par ordenado $SI = (U, A)$, em que U é um conjunto finito e não vazio de elementos, chamado de universo e A é um conjunto finito e não vazio de atributos, tal que, $a : U \rightarrow V_a$ para todo $a \in A$. O conjunto V_a é chamado de conjunto de valores do atributo a . Os elementos que formam o conjunto A são chamados de atributos condicionais, os quais representam as ocorrências observadas nos atributos.

A Tabela 4.1, apresenta um exemplo de sistema de informação definido pelo seu conjunto universo $U = \{o_1, o_2, o_3, o_4, o_5\}$, pelos seus atributos condicionais $A = \{Estação, Temperatura, Vento\}$, e pelos conjuntos de valores possíveis dos atributos $V_{Estação} = \{outono, inverno, primavera, verão\}$, $V_{Temperatura} = \{baixa, média, alta\}$ e $V_{Vento} = \{forte, moderado\}$.

Tabela 4.1 - Sistema de Informação.

U	Atributos Condicionais		
	Estação	Temperatura	Vento
o_1	Outono	média	forte
o_2	Inverno	baixa	moderado
o_3	Primavera	alta	moderado
o_4	Verão	alta	forte
o_5	Outono	média	forte

As linhas $\{o_1, o_2, o_3, o_4, o_5\}$ são chamadas de exemplos ou objetos. As propriedades dos exemplos são determinadas através dos valores atribuídos aos atributos condicionais.

4.3 Sistemas de Decisão

Um Sistema de Decisão (SD) é um SI considerando um atributo de decisão d que não pertence ao conjunto de atributos A . Formalmente, um sistema de decisão é dado por: $SD = (U, A\{d\})$, em que $d \notin A$ é o atributo de decisão (KOMOROWSKI et al., 1998).

Na Tabela 4.2 é apresentado um sistema de decisão, que também será usado ao longo deste capítulo para exemplificar os conceitos apresentados da TCA. Neste sistema, o atributo de decisão *Chuva* é acrescentado, referindo-se à ocorrência de chuva, com

conjunto de valores $V_{Chuva} = \{sim, não\}$.

Tabela 4.2 - Sistema de Decisão.

U	Atributos Condicionais			Atributo de decisão
	Estação	Temperatura	Vento	Chuva
o_1	Outono	média	forte	Sim
o_2	Inverno	baixa	moderado	Não
o_3	Primavera	alta	moderado	Sim
o_4	Verão	alta	forte	Sim
o_5	Outono	média	forte	Não

4.4 Relação de Indiscernibilidade

Um sistema de decisão expressa todo o conhecimento sobre o modelo. Este sistema, em parte, pode ser desnecessariamente grande, devido à representação redundante de atributos idênticos ou à existência de atributos supérfluos.

A relação de indiscernibilidade (IND) (relação de não-discernimento) é usada para medir a similaridade entre dois ou mais objetos. Se tal relação existe entre os objetos, isso implica que os valores de seus atributos são idênticos e não podem ser discernidos entre si, sendo assim, alguns atributos são definidos como redundantes.

Dado um sistema de informação $SI = (U, A)$, então para qualquer subconjunto de atributos $B \subseteq A$ pode existir uma relação de equivalência $IND_A(B)$ definida por Pawlak (1982):

$$IND_A(B) = \{(x, x') \in U^2 | \forall a \in B, a(x) = a(x')\} \quad (4.1)$$

em que $IND_A(B)$ é a relação de indiscernibilidade B . Se $(x, x') \in IND_A(B)$, então os objetos x e x' são indiscerníveis relativamente a qualquer atributo no conjunto B . A relação de equivalência IND , divide os subconjuntos em classes de equivalência, sendo denotada por $[x]_B$ (KOMOROWSKI et al., 1998).

Considerando o SD apresentado na Tabela 4.2, a partição formulada pela relação de indiscernibilidade produz as seguintes classes de equivalência:

- $IND_{(Estação)} = \{o_1, o_5\}, \{o_2\}, \{o_3\}, \{o_4\}$
- $IND_{(Temperatura)} = \{o_1, o_5\}, \{o_2\}, \{o_3, o_4\}$

- $IND_{(Vento)} = \{o_1, o_4, o_5\}, \{o_2, o_3\}$
- $IND_{(Estação, Temperatura)} = \{o_1, o_5\}, \{o_2\}, \{o_3\}, \{o_4\}$
- $IND_{(Estação, Vento)} = \{o_1, o_5\}, \{o_2\}, \{o_3\}, \{o_4\}$
- $IND_{(Temperatura, Vento)} = \{o_1, o_5\}, \{o_2\}, \{o_3\}, \{o_4\}$
- $IND_{(Estação, Temperatura, Vento)} = \{o_1, o_5\}, \{o_2\}, \{o_3\}, \{o_4\}$

A partir do conceito da relação de indiscernibilidade é possível definir atributos redundantes. Se um conjunto de atributos possuem valores idênticos, então, qualquer atributo que pertence ao subconjunto e não pertence ao conjunto de atributos é redundante.

Considerando o conjunto de atributos $B = \{\text{Estação, Temperatura e Vento}\}$ os exemplos $\{o_1\}$ e $\{o_5\}$ apresentam os mesmos valores, portanto existe uma relação de indiscernibilidade. Enquanto que para o subconjunto $B = \{\text{Temperatura}\}$, os exemplos $\{o_3\}$ e $\{o_4\}$ estão na mesma classe de equivalência, assim como os $\{o_1\}$ e $\{o_5\}$.

Neste contexto, um conjunto de atributos que não possuem atributos redundantes é nomeado de conjunto mínimo (ou independente). Um conjunto R de atributos é um reduto de um outro conjunto P de atributos, se R é mínimo e as relações de indiscernibilidade, definidas em R e P são as idênticas.

A relação de indiscernibilidade é o principal conceito envolvido em TCA, responsável pelo agrupamento de valores que são conceitualmente equivalentes. Cada classe vai conter aqueles objetos que são indiscerníveis entre si, ou seja, aqueles para os quais os valores dos atributos que definem a relação são os mesmos.

No exemplo da [Tabela 4.2](#), o conjunto original de atributos $\{\text{Estação, Temperatura, Vento}\}$, os objetos o_1 e o_5 são caracterizados pelos mesmos valores em todos os atributos, sendo assim, é possível reduzi-los, formando assim a classe C_1 , representada pela [Tabela 4.3](#).

4.5 Aproximação de Conjuntos

Uma relação de equivalência induz um particionamento do universo. Estas partições podem ser usadas para construir novos subconjuntos do universo. Assim, pode-se construir uma aproximação de $B \subseteq A$, utilizando as informações contidas no

Tabela 4.3 - Redução de Informação.

	Atributos Condicionais		
U	Estação	Temperatura	Vento
C_1	Outono	média	forte
C_2	Inverno	baixa	moderado
C_3	Primavera	alta	moderado
C_4	Verão	alta	forte

conjunto de atributos $X \subseteq U$, construindo as aproximações B-inferior e B-superior de X , denotado por $\underline{B}X$ e $\overline{B}X$, respectivamente.

Para o exemplo mostrado na Tabela 4.2, não há uma resposta única para o atributo de decisão {Chuva} para os objetos o_1 e o_5 . Neste exemplo, os atributos condicionais possuem uma relação de indiscernibilidade, por outro lado, os valores para o atributo de decisão são discerníveis, sendo assim, existe uma contradição (conflitantes) tornando essas classes não-determinísticas (KOMOROWSKI et al., 1998).

A aproximação inferior é o conjunto de objetos que podem ser certamente classificados como membros de X . A aproximação inferior de X em relação a B é denotada por $\underline{B}X$:

$$\underline{B}X = U \{X \in U / IND(B) | X \subseteq X\}. \quad (4.2)$$

A aproximação superior são os objetos que podem ser classificados como possíveis membros do conjunto X , ou seja, são tratados pelos conjuntos que apresentam padrões idênticos nos atributos condicionais e valores diferentes no atributo de decisão, gerando decisões contraditórias. A aproximação superior de X em relação a B é denotada por $\overline{B}X$:

$$\overline{B}X = U \{X \in U / IND(B) | X \cap X \neq \emptyset\}. \quad (4.3)$$

O conjunto F , chamado de região de fronteira ou borda de X , é formado a partir dos objetos que não podem ser classificados como pertencentes a X , em que, a diferença de $\underline{B}X$ para $\overline{B}X$, são os objetos que são impossíveis de serem classificados como membros do conjunto X , definida como:

$$F(X) = \underline{B}X - \overline{B}X. \quad (4.4)$$

O conjunto E é chamado de região externa de X , é formado pelos objetos que podem ser classificados como não pertencem a X , na forma:

$$E(X) = U - \overline{B}X. \quad (4.5)$$

Para o exemplo apresentado na [Tabela 4.2](#), as aproximações para as classes do atributo de decisão {chuva}, $X_s = Sim = \{o_1, o_3, o_4\}$ e $X_n = Não = \{o_2, o_5\}$ são:

Tabela 4.4 - Aproximações do Sistema de Decisão.

$\underline{B}(X_s) = \{o_3\}, \{o_4\}$	$\underline{B}(X_n) = \{o_2\}$
$\overline{B}(X_s) = \{o_3\}, \{o_4\}, \{o_1, o_5\}$	$\overline{B}(X_n) = \{o_2\}, \{o_1, o_5\}$
$F(X_s) = \overline{B}(X_s) - \underline{B}(X_s) = \{o_1, o_5\}$	$F(X_n) = \overline{B}(X_n) - \underline{B}(X_n) = \{o_1, o_5\}$
$E(X_s) = U - \overline{B}(X_s) = \{o_2\}$	$E(X_n) = U - \overline{B}(X_n) = \{o_1\}, \{o_3\}, \{o_4\}$

4.6 Matriz de Discernibilidade

Pawlak (1982), Komorowski e Øhrn (1999) propuseram um método para o cálculo de reduções de atributos, a partir de funções produzidas pela matriz de discernibilidade.

A matriz de discernibilidade é construída a partir de comparações entre os valores dos atributos. Assim, os atributos que apresentam valores discerníveis são inseridos na matriz, em que cada entrada consiste em um conjunto de atributos que difere os objetos x_i e x_j . A matriz de discernibilidade do sistema é simétrica de dimensão $n \times n$, em que, cada posição $M_{i,j}$ da matriz, é preenchida por um subconjunto de atributos que discernem os objetos x_i e x_j (KOMOROWSKI; ØHRN, 1999).

$$M_{i,j} = \{a \in B | a(x_i) \neq a(x_j)\}, \quad (4.6)$$

para $i, j = 1, \dots, n$.

Para efeito de simplificação, tem-se as seguintes abreviações dos atributos condicionais: Estação = E , Temperatura = T e Vento = V . A [Tabela 4.5](#) apresenta a matriz de discernibilidade, construída, a partir do SD .

Tabela 4.5 - Matriz de Discernibilidade.

	C_1	C_2	C_3	C_4
C_1	\emptyset			
C_2	E, T, V	\emptyset		
C_3	E, T, V	E, T	\emptyset	
C_4	E, T	E, T, V	E, V	\emptyset

4.7 Função de Discernibilidade

Com a matriz de discernibilidade construída, o passo seguinte é a construção da função de discernibilidade que são as uniões de cada entrada a partir da Tabela 4.5. Em cada célula, as variáveis booleanas são conectadas por meio do operador disjuntivo.

A função de discernibilidade f_A para o sistema de decisão é uma função booleana de m atributos condicionais, a_1^*, \dots, a_m^* (correspondente aos atributos a_1, \dots, a_m), obtida a partir da matriz de discernibilidade, sendo expressa por (KOMOROWSKI; ØHRN, 1999):

$$f_A(a_1^*, \dots, a_m^*) = \bigwedge \left\{ \bigvee m_{i,j}^* \mid 1 \leq j \leq i \leq n, \quad m_{i,j} \neq \emptyset \right\}, \quad (4.7)$$

em que $m_{i,j} = \{a^* \mid a \in m_{i,j}\}$.

A função de discernibilidade obtida a partir da matriz de discernibilidade (Tabela 4.5), é mostrada abaixo, sendo, cada linha da função de discernibilidade correspondente a uma coluna da matriz de discernibilidade.

$$\begin{aligned} f_A(a_1^*, \dots, a_m^*) = & (E \vee T \vee V) \wedge (E \vee T \vee V) \wedge (E \vee T) \wedge \\ & (E \vee T \vee V) \wedge (E \vee T) \wedge (E \vee T \vee V) \wedge \\ & (E \vee T \vee V) \wedge (E \vee T) \wedge (E \vee V) \wedge \\ & (E \vee T) \wedge (E \vee T \vee V) \wedge (E \vee V) \wedge \\ & (E \vee T \vee V) \end{aligned}$$

4.8 Redução

Os dados contidos em um sistema de informação podem ser utilizados para discernir atributos somente até um determinado grau. Contudo, os atributos podem ser desnecessariamente volumosos e, ao mesmo tempo, nem todos são necessários para desempenhar tal tarefa. Sendo assim, a função de discernibilidade é utilizada para realizar o processo de redução de atributos sem alterar a relação de indiscernibilidade.

O processo de redução é feito através dos chamados redutos (RED), que são subconjuntos de atributos com capacidade de representar o conhecimento da base de dados, ou seja, agrupar em classes os objetos que são indiscerníveis, a partir da matriz de discernimento, formada pelos atributos discerníveis. Contudo, apenas os atributos que preservam a relação de indiscernibilidade são mantidos na redução, ou seja, existe um subconjunto de atributos condicionais que mantêm a mesma par-

tição, levando em consideração as classes formadas pelo atributo de decisão. Os atributos restantes são redundantes, desde que suas remoções mantenham a mesma classificação (KOMOROWSKI; ØHRN, 1999).

4.9 Ferramenta Rosetta

O software Rosetta (*Rough Set Toolkit for Analysis of Data*), é um sistema abrangente para a realização de análise de dados no contexto da TCA, os dados são analisados em busca de regras que descrevam o seu comportamento (ØHRN, 1999b). Rosetta foi desenvolvido pelo grupo de lógica do Instituto de Matemática da Universidade de Varsóvia e pelo grupo de Sistemas de Conhecimento da Universidade Norueguesa de Ciência e Tecnologia (NTNU).

A ferramenta Rosetta foi adotada para realizar a redução de dimensionalidade dos dados climáticos, sendo assim, algumas de suas funcionalidades serão melhor descritas para fundamentar as decisões feitas na direção do uso de um algoritmo. Para maiores informações sobre esta ferramenta podem ser encontradas em Øhrn (1999b), Øhrn (1999a).

O sistema Rosetta possui um conjunto de flexíveis e poderosos algoritmos, em um ambiente de fácil utilização, projetado para suportar todas as fases da metodologia de modelagem baseada em discernibilidade. As informações em relação aos dados e as etapas, são apresentadas ao usuário em forma de árvore, o qual favorece a leitura e interpretação das informações contidos em um projeto no sistema. A Figura 4.1 apresenta a interface do software Rosetta com todos os seus algoritmos.

4.9.1 Pré-processamento

A etapa do pré-processamento, compreende a limpeza dos dados, ou seja, a remoção de registros de dados incompletos ou ausentes presentes na base de dados.

Os algoritmos de tratamento de dados incompletos disponíveis no Rosseta são: *Remove Incompletes* (Remoção dos incompletos), *Mean/Mode fill* (Substituição pela média), *Conditioned mean/mode fill* (Substituição pela média condicionada à decisão), *Combinatorial completion* (Substituição combinatorial) e *Conditioned combinatorial completion* (Substituição combinatorial condicionada).

Neste trabalho utilizou-se o algoritmo Remoção dos incompletos (*Remove Incompletes*), para realização do pré-processamento, o qual realiza a remoção de todos os exemplos que possuem um ou mais valores de atributos ausentes. Esta opção

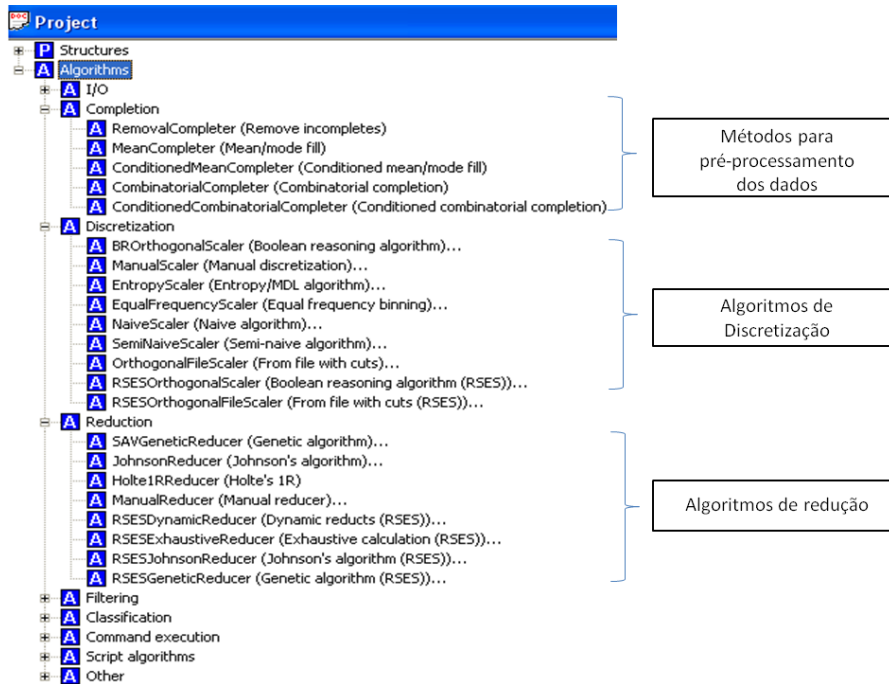


Figura 4.1 - Opções que a ferramenta Rosetta oferece.

deve ser utilizada com cautela, pois a remoção de exemplos com dados ausentes afeta o poder de descrição dos atributos, uma vez que um único valor ausente em um exemplo todos os outros valores dos atributos daquele mesmo exemplo serão desconsiderados (ØHRN, 1999b).

4.9.2 Discretização

A abordagem TCA trabalha apenas com valores discretos, sendo assim, para os atributos numéricos é necessário aplicar o processo de discretização para torná-los em nominais (categóricos). Entretanto quando os atributos numéricos passam por esse processo de discretização ocorre perda de informação, pelo fato dos valores ficarem reduzidos em intervalos, por outro lado, existem muitos algoritmos que conseguem minimizar essa perda de informação (ØHRN, 1999b).

Os algoritmos de discretização disponíveis no sistema Rosetta são: Raciocínio Booleano (*Boolean reasoning algorithm*), Discretização Manual (*Manual discretization*), Entropia (*Entropy/MDL algorithm*), Intervalos Iguais (*Equal frequency binning*), Naive (*Naive algorithm*), Semi-Naive (*Semi-naive algorithm*), Raciocínio Booleano (RSES). Øhrn (1999a) constata que o método de discretização através do Raciocínio Booleano produz resultados eficientes, discretizando os valores contínuos com o menor conjunto de pontos de corte possível.

Em vista do argumento apresentado por (ØHRN, 1999a), foi utilizado o Raciocínio Booleano (*Boolean reasoning algorithm*) no processo de discretização. Este algoritmo utiliza um procedimento de raciocínio booleano para descartar os pontos de corte excessivos. O subconjunto restante é um conjunto de pontos mínimos de cortes que preservam a relação de indiscernibilidade inerente ao sistema de decisão.

4.9.3 Redução de atributos

Uma das principais características da abordagem TCA é a redução de dimensionalidade, as quais são conjuntos mínimos de atributos relevantes que preservam o mesmo conhecimento, ou seja, possuem a mesma relação de indiscernibilidade, representado pelo conjunto de todos os atributos.

O sistema Rosetta possui diferentes métodos de redução, projetados para diminuir a representação da tabela de decisão, sendo eles: Algoritmo Genético (SAVGeneticReducer), Algoritmo de Johnson (JohnsonReducer), o *Holte's 1R*, o *Manual reducer*, o *Dynamic reducts* (RSES) e *Exhaustive calculation* (RSES).

O *SAVGeneticReducer*, implementa um Algoritmo Genético para computação dos conjuntos mínimos (MC), o qual suporta informações de custo e soluções aproximadas (ØHRN, 1999b). Os redutos são recombinados segundo uma função de custo em que os avaliam e os mantém em uma lista e somente aqueles que satisfazem a função de custo são mantidos para serem recombinados (ØHRN, 1999b). A recombinação ocorre enquanto a lista de redutos sofrem mudanças, ou enquanto a qualidade da população aumenta segundo a função custo. Ao final deste processo cada reduto possui um valor de suporte associado como medida de qualidade daquele reduto na representação do conhecimento, preservando assim a relação de indiscernibilidade segundo um fator de credibilidade. A Equação 4.8 representa a função custo do algoritmo genético disponível no sistema Rosetta.

$$f(B) = (1 - \alpha) \times \frac{\text{custo}(A) - \text{custo}(B)}{\text{custo}(A)} + \alpha \times \min \left\{ \epsilon, \frac{|S \text{ em } S| S \cap B \neq \emptyset|}{|S|} \right\} \quad (4.8)$$

em que ϵ é a fração de acertos (*hitting fraction*) ou grau de aproximação para soluções aproximadas. O parâmetro α define uma ponderação entre o conjunto custo e a fração de acerto ϵ . B é um subconjunto de A (conjunto de atributos) e S é o conjunto correspondente à função de discernibilidade. Os subconjuntos B de A são encontrados através da busca da evolução conduzida pela função de custo.

Outro parâmetro relativo as soluções aproximadas é o “extra *keep list*”, denotado

por k , que são listas contendo subconjuntos da função de discernibilidade. Se $k = 0$, então somente uma *keep list* referente à ϵ é retornado. Por outro lado, se $k > 0$ então $k + 1$ grupos de conjuntos de acertos são retornados, em que cada grupo possui uma “fração de acertos” acertando uniformemente espaçados entre ϵ e 1. Se $k = 1$ então o próprio conjunto mínimo de acertos é retornado (ØHRN, 1999b).

Exemplo: Se o usuário especificar soluções aproximadas com os parâmetros $\epsilon = 0.7$ e $k = 3$. São computados 4 grupos de conjuntos mínimos de acerto, com frações de acerto 0.7, 0.8, 0.9 e 1.

Ao final do processo, cada reduto possui um valor de suporte associado como medida de qualidade na representação do conhecimento, sendo assim, neste trabalho foram considerados os redutos que possuem no mínimo 70% ($\epsilon = 0.7$) de fração de acerto.

5 CONFIGURAÇÃO AUTOMÁTICA DE REDES NEURAIIS

Neste capítulo, é apresentada uma formulação para a representação do problema de otimização de parâmetros para topologias de redes neurais supervisionadas. É apresentado também, o processo de otimização da fase de treinamento da rede neural utilizando a meta-heurística MPCA.

5.1 Otimização da arquitetura da rede neural por meta-heurística

O uso de meta-heurísticas tem sido proposto como uma alternativa eficiente para a configuração de parâmetros de topologias de redes neurais (CARVALHO et al., 2011; BENARDOS; VOSNIAKOS, 2007). A definição da arquitetura de uma rede neural pode ser formulada como um problema de otimização, em que cada ponto no espaço de busca representa uma arquitetura.

O desempenho da rede neural é dependente da escolha dos seus parâmetros de treinamento e arquitetura. A definição dos parâmetros depende do problema sendo resolvido. Esse processo tão pouco compreendido é normalmente realizado por experimentação empírica. Entretanto, pequenas diferenças nestes parâmetros podem levar a grandes diferenças tanto no tempo de treinamento como na generalização obtida. A quantidade de neurônios na camada intermediária é primordial no processo de treinamento. Se esse número for exageradamente pequeno, pode ser insuficiente para a rede neural aprender os padrões existentes nos dados, conduzindo a um *underfitting*. Por outro lado, se o número de neurônios for grande, a rede pode memorizar o conjunto de treinamento, perdendo sua capacidade de generalização *overfitting* (BRAGA et al., 2000).

No processo de definição de parâmetros por experimentação empírica, conhecido também como “tentativa e erro”, o especialista modifica os parâmetros da topologia, até alcançar uma solução em que, no seu entendimento, pode ser a solução ideal para o problema em questão. Entretanto, este processo pode não apresentar resultados autênticos, uma vez que não há confiabilidade na obtenção de uma topologia ótima, além do exaustivo treinamento e a qualidade da generalização.

Pode-se dizer que o sucesso desse processo depende da experiência e da intuição do especialista humano, envolvendo, portanto, um alto grau de incerteza. Na experimentação empírica muito tempo de máquina é desperdiçado em simulações desnecessárias e que não abrangem a totalidade das opções de arquitetura disponíveis para um determinado problema.

Assim, neste trabalho foram desenvolvidas duas estratégias distintas para o processo de otimização de parâmetros para topologias de redes neurais supervisionadas: uma formulação mono-objetivo usando o MPCA; e uma formulação multiobjetivo utilizando o NSGA-II.

5.1.1 Solução com otimização mono-objetivo por MPCA

Em um problema de otimização mono-objetivo, uma única medida de desempenho é considerada entre as várias possíveis, resultando em um conjunto exclusivo de parâmetros, o qual fornecerá uma solução ótima correspondente às soluções extremas (mínimas ou máximas) da função objetivo do problema em estudo.

Para a abordagem mono-objetivo, cinco parâmetros de domínio misto de variáveis são otimizados: dois contínuos (o parâmetro de taxa de aprendizagem (η), e a constante *momentum* (α)); e três discretos (número de camadas intermediárias (n_{camada}), número de neurônios em cada camada intermediária ($n_{neurônios}$), e o tipo de função de ativação ($f_{ativação}$)).

5.1.1.1 Função Objetivo

Para o problema de otimização mono-objetivo, a função objetivo J consiste na combinação de dois critérios de erro multiplicado por um fator de penalização, devido à complexidade da arquitetura da RNA. O valor mínimo de J corresponde à arquitetura simples e com um bom desempenho. Esta função foi definida por (CARVALHO et al., 2011) representada na Equação 5.1.

$$J = penalidade \times \frac{\rho_1 E_{train} + \rho_2 E_{gen}}{\rho_1 + \rho_2}, \quad (5.1)$$

em que $\rho_1 = 1$ e $\rho_2 = 0, 1$, os mesmos valores propostos em (CARVALHO et al., 2011). Estes são fatores de ajuste que modificam a relevância atribuída aos erros de treinamento e generalização, respectivamente. Para o caso em que $\rho_1 > \rho_2$ é dada maior importância para a capacidade da rede em memorizar os padrões apresentados, caso contrário, a ênfase será atribuída à capacidade de generalização.

O critério de erro de treinamento (E_{train}) fornece uma indicação quantitativa do nível de treinamento da rede. O critério de erro de generalização (E_{gen}) refere-se à capacidade da rede neural identificar e responder padrões semelhantes baseado no conhecimento aprendido, ou seja, se a rede neural produziu saídas adequadas para entradas não observadas no seu processo de treinamento. O Erro Quadrático Médio

(EQM) é a métrica usada nestes dois critérios (E_{train} e E_{gen}), descrita na Equação 5.2:

$$EQM = \frac{1}{N} \sum_{k=1}^N (d_k - \hat{y}_k)^2, \quad (5.2)$$

em que N é o número de padrões de treinamento, d_k é a saída desejada e \hat{y}_k é a saída produzida pela rede neural.

O termo *penalidade* é uma medida de complexidade computacional da arquitetura de RNA, utilizando o número de neurônios presentes na camada intermediária ($n_{neurônios}$) e o número total de épocas (n_{epocas}) para alcançar a convergência. O objetivo é priorizar topologias menores, evitando o *overfitting*. O termo de penalidade é expresso por:

$$penalidade = C_1 e^{(n_{neurônios})^2} + C_2 (n_{epocas}) + 1, \quad (5.3)$$

em que $C_1 = 5 \times 10^{-8}$ e $C_2 = 5 \times 10^{-5}$, são parâmetros de ajuste para encontrar um equilíbrio nos fatores de medição da complexidade.

Ao lidar com problema de otimização de arquitetura de RNA supervisionada, o objetivo principal é encontrar arquiteturas que apresentem um bom desempenho para o problema em análise e que produzam respostas melhores em relação as saídas fornecidas por uma RNA não-otimizada, ou seja, por experimentação empírica (CARVALHO et al., 2011).

5.1.1.2 Algoritmo MPCA aplicado em otimização de parâmetros

O pseudo-código MPCA para RNAs é apresentado no Algoritmo 1. Cada partícula está estruturada em um vetor de soluções de tamanho D , sendo este a dimensão do problema (número de parâmetros otimizados) e *Fitness* é um número real com o valor da função objetivo.

Assim, os parâmetros que regulam o funcionamento do MPCA são: a variável *AtualP* é a partícula atual; a variável *novaP* é uma nova partícula; *NAFO* é o número de avaliações da função objetivo; a variável *MelhorP* é a melhor partícula em cada iteração e $N_{experimentos}$ é o número total de experimentos. Os parâmetros *LI* e *LS* são os limites inferior e superior, respectivamente, para a intensidade de busca da perturbação local, os parâmetros *LB* e *UB* são vetores que contém o valor mínimo e máximo para cada variável.

Algoritmo 1 MPCA

Divide os dados em subconjuntos de treinamento e validação.

Define os parâmetros de controle do MPCA: número de processadores ($N_{\text{processadores}}$); número de partículas em cada processador ($N_{\text{partículas}}$); disparo para atualização *blackboard* ($N_{\text{blackboard}}$); os limites (LI) e (LS) para perturbação; limites (LB) e (UB) em vetor de soluções.

```
para  $i \leftarrow 1, N_{\text{processadores}}$  faça
  para  $j \leftarrow 1, N_{\text{partículas}}$  faça
    AtualP $_{i,j}$ .Solução = SOLUÇÃOALEATÓRIA
     $\triangleright$  avalia a função objetivo J
    Treina a RNA para AtualP.Solução
    AtualP.Fitness $_{i,j}$  =  $J$ 
  fim para
fim para
para  $i \leftarrow 1, N_{\text{processadores}}$  faça
  MelhorP $_i$  = ATUALIZABLACKBOARD(AtualP $_{i,-}$ )
fim para
NAFO = 0
lastUpdate = 0
enquanto NAFO < NAFO $_{\text{max}}$  faça \triangleright ou outro critério de parada
  para  $i \leftarrow 1, N_{\text{processadores}}$  faça
    para  $j \leftarrow 1, N_{\text{partículas}}$  faça
      novaP $_{i,j}$  = PERTURBAÇÃO(AtualP $_{i,j}$ .Solução)
      se novaP $_{i,j}$ .Fitness < AtualP $_{i,j}$ .Fitness então
        AtualP $_{i,j}$  = novaP $_{i,j}$ 
        AtualP $_{i,j}$  = EXPLORAÇÃO(AtualP $_{i,j}$ )
      senão
        AtualP $_{i,j}$  = ESPALHAMENTO(AtualP $_{i,j}$ , novaP $_{i,j}$ , MelhorP $_i$ )
      fim se
      se AtualP $_{i,j}$ .Fitness < MelhorP $_i$ .Fitness então
        MelhorP $_i$  = AtualP $_{i,j}$ 
      fim se
    fim para
    se NAFO – últimaAtualização >  $N_{\text{blackboard}}$  então
      MelhorP $_i$  = ATUALIZABLACKBOARD(AtualP $_{i,-}$ )
      lastUpdate = NAFO
    fim se
  fim para
fim enquanto
para  $i \leftarrow 1, N_{\text{processadores}}$  faça
  MelhorP $_i$  = ATUALIZABLACKBOARD(AtualP $_{i,-}$ )
fim para
devolve MelhorP $_1$ 
```

5.1.2 Solução com otimização multiobjetivo com NSGA-II

Em problemas de otimização multiobjetivo, o intuito é otimizar mais de uma função objetivo, simultaneamente, resultando em vários conjuntos de parâmetros que fornecem um conjunto de soluções não-dominadas, formando o conjunto Pareto (soluções ótimas). Na presente tese, a resposta desejada é uma topologia ótima, a qual é obtida selecionando o elemento do conjunto de Pareto mais próximo da origem de coordenadas.

O problema mono-objetivo apresentado na Subseção 5.1.1, é formulado como um problema multiobjetivo, transformando a função objetivo (Equação 5.1) em duas funções conflitantes: uma para medir o desempenho da rede neural (Equação 5.4), e a outra para medir a complexidade da RNA (Equação 5.5):

$$J_1 = \frac{\rho_1 E_{train} + \rho_2 E_{gen}}{\rho_1 + \rho_2}, \quad (5.4)$$

$$J_2 = C_1 e^{(n_{neurônios})^2} + C_2 (n_{épocas}) + 1. \quad (5.5)$$

As funções objetivo têm diferentes ordens de magnitude. Portanto, a clássica distância euclidiana não é uma boa métrica para selecionar uma entrada do conjunto de Pareto. A distância de origem é calculada por:

$$d = J_2 + \log J_1. \quad (5.6)$$

No problema multiobjetivo, quatro parâmetros de domínio misto de variáveis são otimizados: dois contínuos, o parâmetro de taxa de aprendizagem (η) e a constante *momentum* (α); e dois discretos, número de neurônios na camada intermediária ($n_{neurônios}$) e o tipo de função de ativação ($f_{ativação}$).

5.1.2.1 Algoritmo NSGA-II aplicado em otimização de parâmetros

O Algoritmo 2 ilustra o funcionamento do NSGA-II implementado para otimização de parâmetros da rede neural. Inicialmente, uma população combinada $R_g = P_g + Q_g$ é formada, em que P_g representa a população atual (de pais), e Q_g representa a população de descendentes (filhos) criada a partir da aplicação dos operadores genéticos de seleção, cruzamento (μ_c) e mutação (μ_m).

Os parâmetros do algoritmo NSGA-II são compostos de: número de experimentos $N_{experimentos}$; número de indivíduos na população ($N_{indivíduos}$); D é a dimensão do problema (número de parâmetros que serão otimizados); os operadores de cruzamento

μ_c e mutação μ_m ; tamanho de torneio (*torneio*); e o subconjunto de indivíduos (do inglês *mating pool*) representado por ($sub_{\text{indivíduos}}$) (SESHADRI, 2006)).

Algoritmo 2 NSGA-II

```

1: para  $i \leftarrow 1, N_{\text{indivíduos}}$  faça
2:    $P_{0_i} = \text{SOLUÇÃOALEATÓRIA}$ 
    $\triangleright$  Avalia as duas funções objetivo  $J_1$  e  $J_2$ 
3:   Treina a RNA para  $P_{0_i}$ 
4: fim para
    $\triangleright$  Obter as frentes não-dominadas
5:  $P_0 = \text{ORDENAÇÃOÑÃODOMINÂNCIA}(P_0)$ 
6:  $g = 0$ 
7: enquanto  $g < g_{\text{max}}$  faça  $\triangleright$ ou outro critério de parada
8:    $P_{\text{intermediário}} = \text{TORNEIOBINARIO}(P_g)$ 
9:    $Q_g = \text{CRUZAMENTO\&MUTAÇÃO}(P_{\text{intermediário}})$ 
    $\triangleright$  Avalia as duas funções objetivo  $J_1$  e  $J_2$ 
10:  para  $i \leftarrow 1, N_{\text{indivíduos}}$  faça
11:    Treina a RNA para  $Q_g$ 
12:  fim para
13:   $R_g = P_g \cup Q_g$ 
    $\triangleright$  Obter as frentes não-dominadas
14:   $R_g = \text{ORDENAÇÃOÑÃODOMINÂNCIA}(R_g)$ 
    $\triangleright$  Seleciona as melhores soluções
15:   $P_{g+1} = \text{SELECIONAMELHORES SOLUÇÕES}(R_g)$ 
16:   $g = g + 1$ 
17: fim enquanto
18: devolve  $P_g$   $\triangleright$ Seleciona  $F_1$  como o conjunto solução ótima

```

As etapas de funcionamento do NSGA-II apresentada no Algoritmo 2 são resumidas a seguir:

- Separar os dados de entrada em subconjuntos para o treinamento e validação.
- Definir o número máximo de gerações g e o número de cromossomos (indivíduos) $N_{\text{indivíduos}}$ em cada população. Cada código de cromossomo, possui uma solução D (genes), em que, D é o número total de parâmetros da topologia de uma RNA, sendo assim, cada cromossomo representa uma RNA. Inicializar aleatoriamente a primeira população P_g de tamanho $N_{\text{indivíduos}}$ – linha 1.
- Avaliar as funções objetivos J_1 e J_2 para os $N_{\text{indivíduos}}$ de RNAs – linha 2.
- Classificar as soluções candidatas (vetor de D) na população P_g , executando o algoritmo de ordenação por não-dominância, em relação aos pa-

res de valores objetivos de J_1 e J_2 , e identificar as frentes não-dominadas F_1, F_2, \dots, F_k , sendo F_1 a melhor frente, F_2 a segunda melhor da frente e F_k é o pior da frente – linha 5.

- Aplicar em P_g a seleção de torneio binário com base na *crowding distance*, para gerar uma população intermediária $P_{intermediária}$ de tamanho $N_{indivíduos}$ – linha 8.
- Aplicar os operadores de cruzamento e mutação em $P_{intermediária}$, para criar a população descendentes Q_g de tamanho $N_{indivíduos}$ – linha 9.
- Avaliar as funções objetivos em relação as soluções Q_g – linha 11.
- Se o número máximo de gerações é alcançado, parar e voltar P_g Selecione a primeira frente de Pareto F_1 como o conjunto solução ótima.
- Combinar P_g e Q_g para obter uma população de união $R_g = P_g \cup Q_g$ – linha 13.
- Executar o algoritmo de ordenação por não-dominância em R_g para obter uma população união ordenada – linha 14.
- Selecionar as melhores soluções $N_{indivíduos}$ da união ordenada para criar a próxima população pai P_{b+1} – linha 15.
- Finalmente, a melhor frente em termos de *ranking* de não-dominância e diversidade das soluções individuais é escolhida. Uma vez que a melhor frente é escolhida, o teste da RNA treinada com os parâmetros de topologia ótima é realizada utilizando os dados do conjunto de validação.

5.1.3 Testes preliminares com MPCA e NSGA-II

O método proposto foi validado através de testes preliminares a partir de redes neurais treinadas para fazer assimilação de dados emulando o Filtro de Kalman. Este estudo de caso permitiu a obtenção de alguns resultados preliminares sobre a aplicabilidade e viabilidade do uso da abordagem de configuração automática de parâmetros de topologias de RNAs.

Assimilação de dados é uma ferramenta que combina de modo eficiente dados de observação com dados de um modelo físico-matemático para a determinação do dado de análise, que é usado para executar o modelo de previsão. Este processo é fundamental, por exemplo, na prática operacional da previsão numérica do tempo

e em circulação oceânica. CAMPOS VELHO et al. (2007) definem o processo de assimilação de dados como a ciência de se ter uma adequada combinação de dados de um modelo matemático de previsão com dados de observação, para se determinar os dados de análise.

A PNT (Previsão Numérica do Tempo) tornou-se um recurso significativo de previsão do tempo, sistemas de suporte a decisões e outras diversas aplicações que dependem de informações meteorológicas. Em suma, o modelo de PNT é um conjunto sofisticado de programas de computador que contêm equações/algoritmos matemáticos e físicos que descrevem a atmosfera e a forma como ela muda com o passar do tempo, produzindo as previsões meteorológicas.

A condição inicial de uma previsão PNT vem da análise, que se constitui na assimilação de diversas observações meteorológicas dentro do modelo. Assimilação de dados, portanto, é um dos temas mais importantes de pesquisa em meteorologia da previsão operacional do tempo. Prover melhorias na qualidade e rapidez no cálculo do dado de análise, resulta no aprimoramento da PNT.

Deste modo, um desafio atual da assimilação de dados consiste em conceber formas para utilizar uma quantidade crescente de dados observados associado com o refinamento de malha nos modelos (aumento do número de pontos na grade computacional), para produzir uma análise dentro do tempo disponível na prática operacional. Assim, a pesquisa usando redes neurais para assimilação é uma tentativa de solução para este desafio (ver os trabalhos de Nowosad et al. (2000); Härter e CAMPOS VELHO (2005); Cintra (2010); e Furtado (2012)).

Matematicamente a assimilação de dados é um processo de dois passos:

(i) Passo da previsão

$$\eta_n^p = M(\eta_{n-1}^a), \quad (5.7)$$

(ii) Passo da análise

$$\eta_n^a = \eta_n^p + \rho, \quad (5.8)$$

em que η_n^p é o vetor de variáveis de estado previsto do modelo, os sobrescritos p e a representam os passos de previsão e análise, respectivamente. M representa o modelo numérico, ρ é o incremento de análise (ou inovação) – esta função é determinada de acordo com a técnica de assimilação de dados, η_n^a são os dados de análise (ou condição inicial (ci)) e n o tempo discreto.

O método desenvolvido por Kalman (1960), o denominado Filtro de Kalman, é uma ferramenta importante na teoria de estimação, que foi aplicado em modelos lineares discretos com observações linearmente relacionadas ao estado do modelo.

5.1.3.1 Filtro de Kalman

Filtro de Kalman (FK) é uma abordagem de estimativa probabilística (KALMAN, 1960; KALMAN; BUCY, 1961; JAZWINSKI, 1970), e tem sido aplicado como método de assimilação de dados (KALNAY, 2003). Esta abordagem assume uma função de densidade probabilística Gaussiana para todas as variáveis aleatórias em um dado processo estocástico.

Para um sistema dinâmico estocástico gaussiano, FK fornece uma estimativa ótima de dados observacionais. A previsão de um modelo matemático é atualizado usando o *ganho de Kalman* para o cálculo da *análise*. As incertezas são calculadas a partir da matriz de covariância. Quando novas observações estiverem disponíveis, o ciclo de assimilação é reiniciado.

A fase de previsão é obtida a partir de uma análise obtida quando novas observações do ciclo estão disponíveis, sendo assim, a análise se propaga com o modelo matemático para representar os fenômenos físicos:

$$\eta_{m+1}^p = M(\eta_m^a), \quad (5.9)$$

O erro de modelagem é estimado pela matriz de covariância do erro de previsão:

$$P_{n+1}^p = M_n P_n^a M_n^T + Q_n, \quad (5.10)$$

em que Q_n é a matriz de covariância do erro de modelagem.

$$\eta_{n+1}^a = \eta_{n+1}^p + K_{n+1}^p \left[\eta_{n+1}^o - H_{n+1} (\eta_{n+1}^p) \right], \quad (5.11)$$

em que η_{n+1}^o é o vetor de dados de observação, a matriz de pesos K (ganho de Kalman) depende das matrizes de covariância de erros e H_{n+1} representa o sistema de observação. A matriz K é calculada por:

$$K_{n+1} = P_{n+1}^p H_{n+1}^T \left[R_{n+1} + H_{n+1} P_{n+1}^p H_{n+1}^T \right]^{-1}, \quad (5.12)$$

sendo R_{n+1} a matriz de covariância dos erros de observação. P^p é a matriz de covari-

ância de erro do modelo e H representa a matriz de um sistema linear de observação. O erro de modelagem associado à análise é estimado por:

$$P_{n+1}^a = [I - K_{n+1}H_{n+1}^T] P_{n+1}^p, \quad (5.13)$$

em que I é a matriz de identidade.

5.1.3.2 Equação da onda 1D linear

O sistema dinâmico usado nos testes é um dos modelos utilizados por Bennett (2002) para avaliação sobre os regimes de assimilação de dados. O modelo matemático é a equação da onda 1D linear de primeira ordem:

$$\frac{\partial \eta}{\partial t} + c \frac{\partial \eta}{\partial x} = F(x, t), \quad (5.14)$$

em que η é o deslocamento, c é a constante de velocidade da fase, F é uma forçante externa, t é o tempo e x é o espaço. F indica a solução avançada, ou seja, a solução *a priori*.

O modelo de onda foi integrado com o método de diferenças finitas centradas no espaço e com o esquema de Crank-Nicholson no tempo. A condição inicial é dado pela seguinte expressão:

$$\eta_F(x, 0) = \eta_0 \frac{1}{\cosh^2[(x - v)/\Delta]} \quad 0 \leq x \leq L_x \quad (5.15)$$

em que $\eta_0 = -60$, $v = c + (\alpha\eta_0)/3$, $\alpha = -1,62 \times 10^{-2}$, $c = 2,42ms$, e $\Delta = 1340$ (BENNETT, 2002).

O modelo foi executado considerando $t_{max} = N_t \times \Delta t = 3000 \times 10s = 30000$ s, e $L_x = N_x \times \Delta x = 128 \times 150m = 19200$ m (FURTADO et al., 2011).

5.1.3.3 Resultados preliminares

Foram realizados testes preliminares para validar a abordagem proposta de configuração automática de parâmetros de RNAs, a qual foi aplicada para emular o processo de assimilação de dados produzido pelo filtro de Kalman. Os resultados mostram duas diferentes formulações para configuração automática: uma formulação mono-objetivo usando o MPCA, e uma formulação multiobjetivo utilizando o NSGA-II.

A comparação foi realizada entre o resultado com a rede neural configurada por um especialista (FURTADO et al., 2011) e a rede neural configurada automaticamente com uso de meta-heurísticas.

Para a implementação do FK, foram utilizadas os seguintes parâmetros: $Q_n = 0.1I$, $R_n = 0.5I$, $H_n = I$, em que, correspondem à matriz de covariância do erro de modelagem, matriz de covariância do erro de observação e o operador que representa o sistema de observação, respectivamente, e I representa a matriz identidade. As observações foram geradas sinteticamente, e os dados foram gerados a partir da integração do sistema, adicionando-se um ruído gaussiano aleatório com 5% de variância.

Para o método proposto de assimilação de dados utilizando redes neurais, a determinação da análise é descrita pela Equação 5.16 (HÄRTER; CAMPOS VELHO, 2005; HÄRTER; CAMPOS VELHO, 2012):

$$\eta_n^a = F_{RNA}(\eta_n^p, y_n^o), \quad (5.16)$$

em que F_{RNA} representa a RNA utilizada no processo de assimilação. A saída desejada da rede é a estimativa obtida com o Filtro de Kalman.

A Tabela 5.1 apresenta os parâmetros utilizados nos experimentos numéricos.

Tabela 5.1 - Parâmetros de controle e critério de parada: MPCA e NSGA-II.

Algoritmo	Parâmetro	Valor
MPCA	$N_{experimentos}$	25
	$NAFO_{max}$	40000
	$N_{processadores}$	240
	$N_{partículas}$	1
	$N_{blackboard}$	100
	D	4
	LI	0.8
	LS	1.2
NSGA-II	$N_{experimentos}$	25
	$N_{indivíduos}$	200
	g_{max}	200
	$sub_{indivíduos}$	100
	D	4
	$torneio$	2
	μ_c	20
	μ_m	20

As Figuras 5.1, 5.2, e 5.3 são os resultados para a assimilação de dados. As observações são representadas por pontos pretos; o estado verdadeiro é representado pela curva azul; e o perfil de onda obtido a partir da análise por RNA emulando o filtro de Kalman é representado pela curva vermelha. A Figura 5.4 apresenta os erros obtidos entre a observação e as assimilações executadas usando diferentes modelos de redes neurais. A curva em vermelho corresponde o erro obtido por experimentação empírica (NN-Especialista); a curva azul é o erro obtido a partir do modelo MLP-MPCA e a curva em preto representa o erro da rede neural configurada pelo algoritmo NSGA-II.

Analisando os resultados, é evidente que os experimentos reproduzidos usando as meta-heurísticas MPCA e NSGA-II obtiveram melhores resultados em relação a RNA definida pelo especialista, apresentando alta fidelidade com a dinâmica real – ver Figuras 5.2 e 5.3. Entretanto, verifica-se que o modelo MLP-NSGA apresentou um índice de erro menor em relação ao modelo MLP-MPCA (ver Figura 5.4).

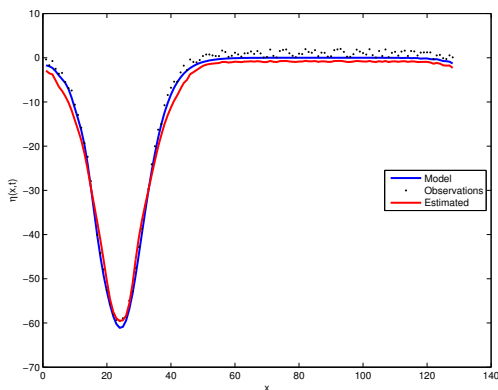


Figura 5.1 - MLP-Especialista

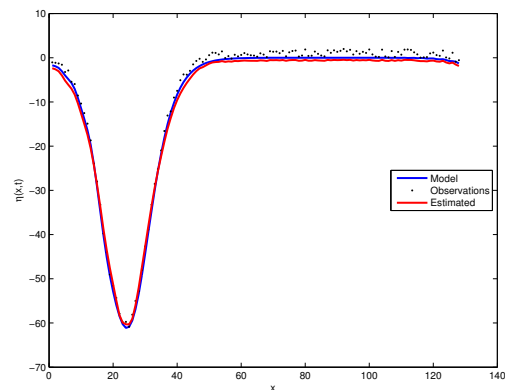


Figura 5.2 - MLP-MPCA

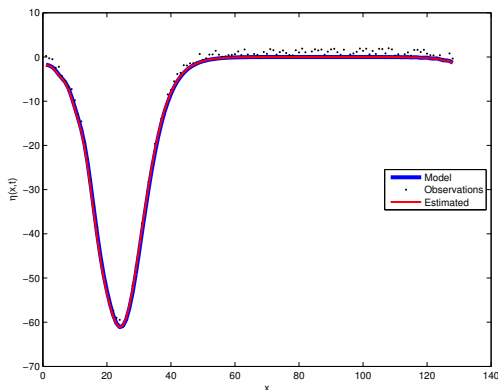


Figura 5.3 - MLP-NSGA-II

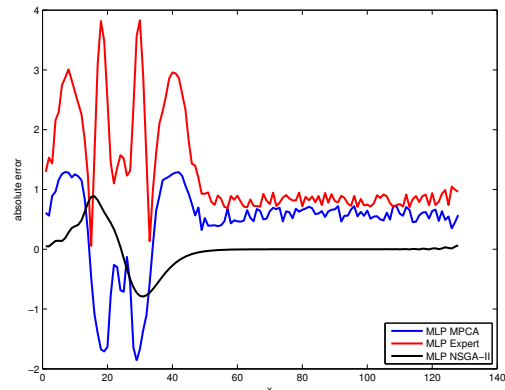


Figura 5.4 - Erro para as três RNAs

A Tabela 5.2 mostra os resultados estatísticos para as arquiteturas encontrados por ambas as meta-heurísticas MPCA e NSGA-II, e a configuração definida empiricamente pelo especialista (FURTADO et al., 2011). São apresentados os melhores, a média e os piores resultados. Verifica-se que o uso de técnicas de otimização para encontrar parâmetros ótimos para topologia de RNA, mostraram-se eficientes para o problema de assimilação de dados, em que reduziu a complexidade e atingiu menor erro em relação ao modelo de rede neural definida por experimentação empírica.

Tabela 5.2 - Parâmetros para arquiteturas de RNA

Método	Estatística	$n_{\text{neurônios}}$	$f_{\text{ativação}}$	η	α	EQM
Empírica	–	3	2	0.9	0.0	1.28E-01
MPCA	Melhor	1	2	0.0172	0.5107	8.29E-02
	Pior	1	2	0.5145	0.5115	8.28E-02
	Média	1	2	0.2267	0.5111	8.29E-02
NSGA-II	Melhor	2	2	0.0114	0.3439	4.00E-04
	Pior	2	2	0.3242	0.9965	5.00E-04
	Média	2	2	0.1749	0.7562	4.00E-04

6 REDES NEURAIIS PARA PREVISÃO CLIMÁTICA DE PRECIPITAÇÃO

Neste capítulo são exibidos os resultados obtidos pelas aplicações das meta-heurísticas MPCA e NSGA-II apresentadas no [Capítulo 3](#) à solução do problema de previsão climática de precipitação. São exibidos também os resultados obtidos com a aplicação da abordagem de redução de variáveis meteorológicas pela Teoria dos Conjuntos Aproximativos apresentada no [Capítulo 4](#).

6.1 Previsão Climática

A previsão climática é definida como um conjunto de condições naturais que dominam uma determinada região, obtida pelo comportamento médio da atmosfera, tais como, a variabilidade e condições extremas do tempo (precipitação, temperatura, vento, entre outros) em um intervalo de tempo suficientemente longo, que são essenciais para caracterizar uma região.

A previsão climática é extremamente essencial para os setores que são afetados pela variabilidade climática, como a produção de energia, agricultura, saúde e outros. Tendo em vista que o Brasil é um país com grande extensão territorial, com diferenciados regimes de precipitação e temperatura, a implementação de modelos para o monitoramento climático já é pauta em pesquisas relacionadas ao tema por [Anochi \(2010\)](#); [Sousa e Sousa \(2010\)](#); [Ruivo et al. \(2015\)](#).

O Modelo de Circulação Geral da Atmosfera (MCGA) em uso no CPTEC tem sua origem no modelo usado para a previsão de tempo de médio prazo pelo NCEP. Esse modelo combinou o código espectral global desenvolvido pelo NCEP e as parametrizações físicas de escala de subgrade desenvolvidas no *Geophysical Fluid Dynamics Laboratory* (GFDL) do NOAA, na Universidade de Princeton, EUA. O CPTEC adotou a versão 1.7 do modelo do *Center for Ocean Land and Atmosphere Studies* (COLA) ([CAVALCANTI et al., 2002](#)).

O modelo MCGA é uma ferramenta central para a previsão do tempo. Este é usado em simulações de clima para estudar variabilidade climática, mudança de clima, previsão sazonal e também para avaliar a sua habilidade para descrever características da atmosfera ([INPE.CPTEC, 2015](#)).

A previsibilidade climática sazonal nos trópicos e extratropicos apresenta sensíveis diferenças. Por exemplo, a região Nordeste do Brasil apresenta alta previsibilidade climática. Essa confiabilidade de previsão, corresponde ao fato que a precipitação

possui um forte acoplamento com as variações de TSM (ROPELEWSKI; HALPERT, 1989). Adicionalmente, a região Sul do Brasil é caracterizada com média previsibilidade. Por outro lado, existem regiões em que a previsibilidade climática é muito baixa. As regiões Sudeste e Centro-Oeste apresentam as menores confiabilidades devido à menor dependência com relação as condições dos oceanos e à grande variedade e variabilidade dos sistemas meteorológicos Zona de Convergência do Atlântico Sul (ZCAS) e Sistemas Frontais (SF) que as atingem (SAMPAIO; SILVA DIAS, 2014).

A ZCAS é definida como uma persistente banda de nebulosidade, estendendo-se desde o sul da região Amazônica até a região central do Atlântico Sul. A ZCAS se caracteriza pelo escoamento de umidade convergente em baixos níveis da troposfera, orientado no sentido noroeste e sudeste. Os SF, ou frentes, são definidos como uma zona de transição entre duas massas de ar de diferentes densidades e temperaturas, que se aproximam e tendem a se manter individualizadas, conservando suas particularidades (VIANELLO; ALVES, 2006).

O modelo empírico de previsão climática proposto nesta tese, foi adaptado para o campo de precipitação para previsão mensal e sazonal, proporcionando a elaboração de cenários futuros que apoiam os estudos de vulnerabilidade. O modelo de RNA recebe como condições iniciais os dados referentes a uma estação e como saída o modelo realiza a previsão climática para uma estação adiante. Um exemplo é apresentado na Figura 6.1, em que o modelo recebe dados referentes à estação outono (março, abril, maio) e terá como saída a previsão de precipitação para a estação inverno (junho, julho, agosto).

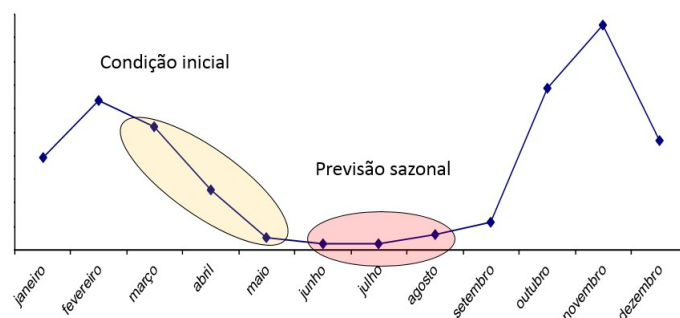


Figura 6.1 - Esquema de previsão sazonal

Fonte: Anochi (2010)

6.2 Ambiente de Teste

A previsão de tempo e clima, realizada por centros de previsão como o Centro de Previsão de Tempo e Clima do Instituto Nacional de Pesquisas Espaciais (CPTEC-INPE), utiliza modelos que procuram descrever o comportamento das condições físico-químicas da atmosfera. Estes são executados em um sistema computacional de grande porte – Tupã.

O Tupã é um sistema de supercomputação composto originalmente por um supercomputador *Cray XT6* capaz de realizar 258 trilhões de cálculos por segundo. O Tupã está entre os mais poderosos supercomputadores para previsão de tempo e estudos sobre as mudanças climáticas para a elaboração de cenários futuros com alta resolução. Em maio de 2012, o sistema foi atualizado para o modelo XE6, com sistema de interconexão *Gemini*, operando em sua capacidade total com 14 gabinetes, 1.304 nós computacionais e 31.296 processadores. O Tupã está instalado nas dependências do CPTEC/INPE (CPTEC, 2015b).

Os experimentos computacionais desta tese, foram executados no Tupã com a seguinte configuração: 240 processadores com 24 núcleos de processamento. O poder computacional disponível foi usado no sentido de possibilitar a execução de vários experimentos simultaneamente, visto que, muitos parâmetros precisam ser ajustados e requerem muitas execuções para validá-los.

6.2.1 Dados Meteorológicos

Para a realização da previsão climática de precipitação, mencionada neste trabalho, foram utilizados dados temporais extraídos da base de dados do NCEP/NCAR (National Centers for Environmental Prediction / The National Center for Atmospheric Research) ¹. *National Oceanic & Atmospheric Administration* (NOAA) é uma importante agência de pesquisa do *National Weather Service* (Serviço Nacional de Meteorologia dos Estados Unidos), responsável pelas pesquisas que envolvem assuntos relacionados ao meio ambiente da Terra, desde a superfície do sol até as profundezas do oceano (NOAA, 2012).

O NCEP/NCAR disponibiliza grandes conjuntos de dados de reanálises, para produzir análises e diagnósticos globais de campos atmosféricos, com o propósito de atender às necessidades das comunidades científicas e monitoramento do clima. Os dados são disponibilizados no formato binário netCDF, através do site

¹<http://www.ncep.noaa.gov/>

[<http://www.esrl.noaa.gov/psd/data/gridded/data.ncep.reanalysis.html>] e abrange o período de 1948 até a atualidade.

As séries históricas de reanálise, geradas a partir de modelos, são construídas com base em séries de variáveis meteorológicas disponíveis a partir de algum tipo observação. Assim, o banco de dados de reanálise pode ser gerado a partir da assimilação de observações climáticas, provenientes de diferentes fontes, como aviões, navios, imagens de satélite, imagens de radar e radiossondagens (KALNAY et al., 1996).

Os dados coletados para os experimentos de previsão de precipitação são mensais e abrangem o período de janeiro de 1990 e fevereiro de 2015 (302 meses). Trata-se de um conjunto de dados globais de reanálise com resolução espacial (latitude e longitude) de $2.5^\circ \times 2.5^\circ$.

As áreas de estudo para realização dos experimentos abrangem três regiões do Brasil:

- a) Nordeste (NE) – latitude 17°S , 0° e longitude 47°W , 35°W ;
- b) Sul (S) – latitude 35°S , 25°S e longitude 60°W , 40°W ;
- c) Sudeste (SE) – latitude 25°S , 15°S e longitude 52°W , 37°W .

A Tabela 6.1 traz uma breve descrição das variáveis meteorológicas selecionadas do NCEP/NCAR, usadas neste estudo, com suas respectivas unidades e níveis (superfície ou níveis de pressão em hectopascal (hPa)).

Tabela 6.1 - Variáveis selecionadas do NCEP/NCAR

Variável	Descrição	Nível
u300	Vento Zonal [m/s]	300 hPa
u500	Vento Zonal [m/s]	500 hPa
u850	Vento Zonal [m/s]	850 hPa
v300	Vento Meridional [m/s]	300 hPa
v500	Vento Meridional [m/s]	500 hPa
v850	Vento Meridional [m/s]	850 hPa
temp	Temperatura [$^\circ\text{C}$]	850 hPa
air	Temperatura na superfície [$^\circ\text{C}$]	superfície
shum	Umidade Específica [g/kg]	500 hPa
prec	Precipitação [mm/dia]	superfície

6.2.2 Visualização de dados

Para a visualização e análise dos resultados, utilizou-se a ferramenta GrADS (*Grid Analysis and Display System*). Este é um software que fornece um ambiente integrado para acesso, manipulação e exibição de dados em ponto de grade. O GrADS utiliza dois arquivos para realizar a visualização: um arquivo descritor (ctl), o qual descreve as características e informações dos arquivos de dados, tais como, as coordenadas geográficas (latitude e longitude), tempo, nível, o número e o nome das variáveis; e um arquivo de dados no formato binário (DOTY, 2009).

Nesta pesquisa, as previsões climáticas obtidas a partir de modelos de redes neurais, encontram-se em arquivos de texto ASCII, com informações sobre as coordenadas geográficas, o tempo e as respectivas previsões de precipitação. Entretanto, para integrar estes arquivos texto ao ambiente GrADS, requerem um pré-processamento. Sendo assim, estes arquivos de texto foram transformados em arquivos binários, a partir de um programa desenvolvido usando a linguagem FORTRAN.

6.2.3 Conjunto de dados para treinamento e generalização

Durante a fase de treinamento, a capacidade de generalização é avaliada através de um conjunto de dados de validação. Os dados de treinamento são divididos em dois subconjuntos, um de exemplos e outro para a validação cruzada. O subconjunto de exemplos é usado no treinamento da rede. O treinamento é interrompido periodicamente após um número determinado de épocas, e a rede é testada com o subconjunto de validação (processo de validação cruzada). Estes ciclos de testes são repetidos até que a convergência ou o número máximo de épocas sejam alcançados. Na fase de generalização, um conjunto não utilizado na fase de treinamento é apresentado à rede para avaliação e verificação de seu desempenho.

Desta perspectiva, o conjunto de dados foi dividido em três grupos:

- Conjunto de treinamento: janeiro de 1990 - dezembro de 2010;
- Conjunto de validação cruzada: janeiro - dezembro de 2011;
- Conjunto de generalização: janeiro de 2012 - fevereiro de 2015.

6.2.4 Redução dos dados meteorológicos com TCA

Identificar, com razoável antecedência e confiabilidade, condições favoráveis à ocorrência de eventos climáticos e realizar em tempo hábil para que se possa tomar as

devidas medidas e conseqüentemente reduzir o impacto negativo de eventos meteorológicos, é algo essencial para a sociedade. Em Meteorologia, dados de diversas fontes, ditos multivariados, são usados, tais como dados de satélites, estações de superfície, boias oceânicas, radiossondas, modelos numéricos de previsão do tempo, dados de ocorrência de descargas elétricas, radar, dentre outros, são usados para previsões de tempo e clima.

Entretanto, as previsões de eventos meteorológicos são complexas pela necessidade de análise de grande volume de dados. Desta forma, propõe-se o uso de uma técnica de mineração de dados, para identificar as variáveis mais significativas para o processo de previsão climática.

Na fase de redução de dimensionalidade dos dados meteorológicos utilizou-se os dados obtidos por Anochi (2010). Para calcular as variáveis mais significativas para o processo de previsão climática, utilizou-se a ferramenta Rosetta, em que os dados disponíveis são pré-processados para extrair informações relevantes (em uma abordagem de mineração de dados).

Primeiramente os dados são discretizados, em seguida são submetidos ao algoritmo de redução para o cálculo dos conjuntos candidatos aproximados, como descrito na Subseção 4.9.3. O principal parâmetro é a fração de acerto (ϵ), cujo valor neste trabalho foi de 0.7, ou seja, todos os atributos que pertencem aos conjuntos candidatos aproximados, que neste caso é a redução, que tem no mínimo 70% de presença na função de discernimento (f_A – apresentado na Seção 4.7). Esses são apresentados como os redutos desejados para a entrada do modelo de previsão baseado em redes neurais.

No modelo de previsão de precipitação usando abordagem de redução de dados, utilizou-se a configuração ótima de parâmetros obtidos pela meta-heurística MPCA.

6.3 Resultados com redes autoconfiguradas

Nesta Seção são apresentados os resultados de previsão de precipitação usando a abordagem de otimização de parâmetros. A identificação da melhor topologia é tratado como um problema de otimização, em que cada ponto no espaço de busca representa uma arquitetura com diferentes parâmetros e pesos.

A variável precipitação é considerada uma das mais importantes variáveis meteorológicas, fornecida por modelos de previsão de tempo e clima. Neste trabalho, a previsão climática sazonal de precipitação é realizada utilizando a rede supervisio-

nada perceptron múltiplas camadas (NN-MLP) como modelo preditivo da variável precipitação.

O mapa de erro para a previsão gerada a partir do uso de redes neurais é calculado por:

$$\text{error} = |y_k - d_k^m|, \quad (6.1)$$

em que y_k é a medida de precipitação e d_k^m é a precipitação calculada pela rede neural configurada pela estratégia m ($m = \text{Empírica, MPCA, NSGA-II}$).

Para a realização da previsão climática de precipitação, o problema de otimização utilizou duas estratégias distintas: um problema de minimização mono-objetivo usando a meta-heurística MPCA e um problema de minimização multiobjetivo utilizando o NSGA-II. Como essas meta-heurísticas são abordagens estocásticas vários experimentos serão executados, a fim de encontrar suficientes soluções representativas. Para o funcionamento dos algoritmos MPCA e NSGA-II, um conjunto de parâmetros precisam ser definidos. A Tabela 6.2 mostra os parâmetros utilizados para a realização dos experimentos numéricos.

Tabela 6.2 - Parâmetros de controle e critério de parada: MPCA e NSGA-II

Algoritmo	Parâmetro	Valor
MPCA	$N_{\text{experimentos}}$	15
	$NAFO$	4000
	$N_{\text{processadores}}$	240
	$N_{\text{particulas}}$	1
	$N_{\text{blackboard}}$	100
	LI	0.8
	LS	1.2
NSGA-II	$N_{\text{experimentos}}$	15
	$N_{\text{individuos}}$	200
	g_{max}	200
	$sub_{\text{indivíduos}}$	100
	$torneio$	2
	μ_c	20
	μ_m	20

A meta-heurística MPCA foi empregada para otimizar cinco parâmetros: número de camadas intermediárias (n_{camada}), número de neurônios em cada camada intermediária ($n_{\text{neurônios}}$), taxa de aprendizagem (η), constante *momentum* (α) e o tipo de função de ativação ($f_{\text{ativação}}$). Por outro lado, na formulação multiobjetivo, a meta-

heurística NSGA-II trabalhou com a otimização de quatro parâmetros: número de neurônios em cada camada intermediária, taxa de aprendizagem, constante *momentum* e o tipo de função de ativação. Os valores permitidos para estes parâmetros são apresentados na Tabela 6.3.

Tabela 6.3 - Parâmetros que definem arquiteturas de RNAs

Tipo	Parâmetros	Valor
Discreto	n_{camadas}	[1, 3]
	$n_{\text{neurônios}}$	[1, 40]
	$f_{\text{ativação}}$	tangente (1) sigmoide (2) gaussiana (3)
Contínuo	η	[0.01, 1.0]
	α	[0.01, 1.0]

6.3.1 Previsão Sazonal

6.3.1.1 Região Nordeste

Para a realização de previsão de precipitação na escala sazonal, toda a região Nordeste do Brasil foi selecionada. As coordenadas geográficas que delimitam a região considerada estão compreendidas entre as longitudes 47°W, 35°W e latitudes 17°S, 0°, como ilustrado na Figura 6.2 (retângulo vermelho no mapa do Brasil).

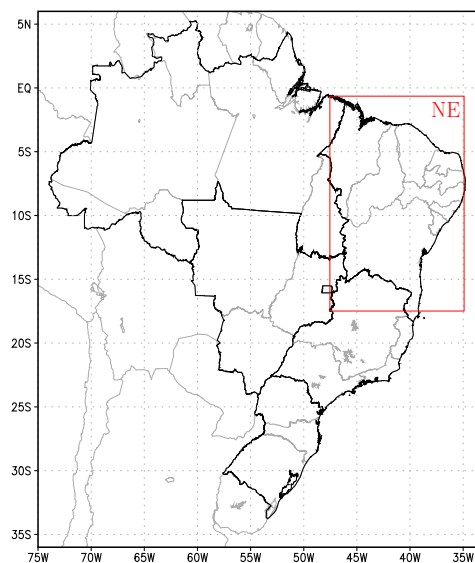


Figura 6.2 - Área de estudo: Nordeste (NE)

Levando-se em conta o regime de precipitação, encontra-se sobre a região Nordeste do Brasil uma alta variedade climática, podendo-se verificar desde o clima semi-árido no interior da região, com precipitação acumulada inferior a 500mm/ano, até o clima chuvoso, observado principalmente na costa leste da região, com precipitação acumulada anual superior a 1500mm (KOUSKY; CHU, 1978). Historicamente a região Nordeste sempre foi afetada por grandes secas ou grandes cheias.

Os parâmetros correspondentes às melhores e piores soluções encontradas após um número consecutivo pré-determinado de avaliações da função objetivo (NAFO) são exibidos na Tabela 6.4 para arquiteturas de NNs identificadas por MPCA e NSGA-II. Comparando os resultados, a topologia utilizada pela meta-heurística NSGA-II foi melhor que o algoritmo MPCA, em termos de erro.

A estratégia de treinamento automático (configuração de parâmetros) utilizando as meta-heurísticas MPCA e NSGA-II, elimina do especialista a responsabilidade de definir uma topologia ótima para a rede neural. Neste experimento, o processo de otimização tem início com uma topologia mínima de uma NN válida e durante o processo de otimização, os parâmetros são alterados.

Tabela 6.4 - Resultados para a previsão climática na região Nordeste

Região	Método	Estatística	$n_{\text{neurônios}}$	$f_{\text{ativação}}$	η	α	EQM
Nordeste	Empírica	–	15	1	0.4	0.9	1.70E-04
	MPCA	Melhor	5	1	0.8534	0.9650	3.01E-08
		Pior	2	1	0.3438	0.0493	5.70E-04
		Média	2	2	0.3515	0.4123	2.70E-03
	NSGA-II	Melhor	2	2	0.6449	0.7609	8.10E-04
		Pior	2	2	0.9565	0.9910	8.90E-04
		Média	2	2	0.5494	0.5682	3.70E-04

Redução de variáveis meteorológicas

As variáveis mais significativas selecionadas pela TCA para a região NE são mostradas na Tabela 6.5. Verifica-se que dentre as dez variáveis selecionadas da base de dados do NCEP/NCAR, conforme apresentado na Tabela 6.1, na Subseção 6.2.1, após o pré-processamento pela TCA (em uma abordagem de mineração de dados) o conjunto de dados reduziu para cinco variáveis (ANOCHI, 2010).

Tabela 6.5 - Variáveis extraídas pela TCA

Variável	% (f_A)
temp	75
u500	76
v300	74
v500	77
shum	71

Precipitação: Verão de 2014

A Figura 6.3(a) apresenta o campo de precipitação de reanálise para a estação Verão de 2014. A Figura 6.3(b) mostra a previsão obtida com a rede treinada de forma empírica (NN-Empírica). As Figuras 6.3(c) e 6.3(d) são os resultados de precipitação usando a configuração ótima determinada pelas meta-heurísticas MPCA (NN:MPCA) e NSGA-II (NN:NSGA), respectivamente. A Figura 6.3(e) exhibe o resultado de previsão usando a NN treinada com o conjunto reduzido de variáveis (TCA:MPCA) e configurada com os parâmetros ótimos obtidos por MPCA. As Figuras 6.3(f), 6.3(g), 6.3(h) e 6.3(i) mostram os mapas de erro para os diferentes modelos de previsão: NN:Empírica, NN:MPCA, NN:NSGA e TCA:MPCA, respectivamente.

Os resultados de previsão sazonal obtidos a partir dos modelos de NNs, foram comparados com a resposta esperada, a qual está disponível nos dados de observação, extraído da base de dados de reanálise do NCEP/NCAR. Verifica-se que a previsão obtida com a rede TCA:MPCA, a qual utilizou o conjunto reduzido de variáveis e foi configurada com os parâmetros ótimos definidos pelo MPCA, identificou o evento de precipitação apresentado no campo observado, com uma margem de erro baixa – ver Figura 6.3(i). Constata-se que a margem de erro do modelo preditivo TCA:MPCA é inferior em relação aos modelos que utilizaram todas as variáveis disponíveis na base de dados no treinamento (NN:Empírica, NN:MPCA e NN:NSGA) – ver Figuras 6.3(f), 6.3(g) e 6.3(h).

Destaca-se, que o modelo de rede TCA:MPCA detectou o evento de precipitação em todos os estados da região NE, especialmente, na Bahia, área em que os modelos NN:Empírica, NN:MPCA e NN:NSGA (usando todas as variáveis) não conseguiram identificar.

Precipitação: Outono de 2014

O campo de reanálise da variável de precipitação para a estação Outono em 2014 é mostrado na Figura 6.4(a). Os resultados de previsão obtidos a partir dos modelos de NNs, são mostrados nas Figuras 6.4(b), 6.4(c), 6.4(d) e, 6.4(e) considerando diferentes abordagens de configuração: NN:Empírica, as estratégias mono-objetiva NN:MPCA, multi-objetivo NN:NSGA e redução de variáveis TCA:MPCA. As diferenças entre a observação e a previsão obtida a partir de NN são apresentados nas Figuras 6.4(f), 6.4(g), 6.4(h) e 6.4(i), com diferentes NNs, NN:Empírica, NN:MPCA, NN:NSGA e TCA:MPCA, respectivamente.

Analisando os resultados de previsão para estação Outono apresentados na Figura 6.4, verifica-se que o modelo TCA:MPCA conseguiu identificar o nível de precipitação ocorrido no estado do Bahia, em que tal característica está mais perto da observação, em relação aos modelos que foram treinados com todo o conjunto de variáveis.

Precipitação: Inverno de 2014

O campo de precipitação de reanálise para a estação Inverno em 2014 é mostrado na Figura 6.5(a). As Figuras 6.5(b), 6.5(c), 6.5(d) e 6.5(e), apresentam os resultados de previsão de precipitação obtidos por redes neurais configuradas de diferentes abordagens: experimentação empírica (NN:Empírica), mono-objetiva (NN:MPCA), multi-objetivo (NN:NSGA) e redução de variáveis configurada com parâmetros ótimos do MPCA (TCA:MPCA), respectivamente. As Figuras 6.5(f), 6.5(g), 6.5(h) e 6.5(i), mostram os mapas de erros para as quatro abordagens de previsão.

Diante dos resultados de previsão apresentados na Figura 6.5, verifica-se que embora o modelo NN:Empírica tenha obtido um bom resultado de previsão nos estados de Maranhão, Piauí e Ceará, com padrões de precipitação semelhantes com a observação, foi o resultado apresentado pela rede TCA:MPCA que identificou os níveis de precipitação em toda a região, especialmente, nos estados costeiros. O mapa de erro apresentado na Figura 6.5(i), apresenta uma margem de erro inferior em relação aos modelos que utilizaram todas as variáveis no treinamento.

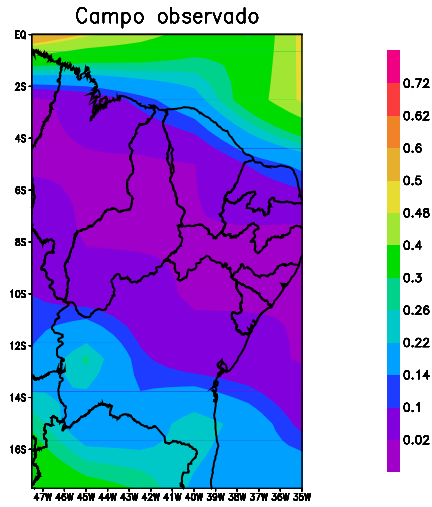
Precipitação: Primavera de 2014

A precipitação de reanálise observada durante a estação Primavera no ano de 2014 é apresentada na Figura 6.6(a). As Figuras 6.6(b), 6.6(c), 6.6(d) e 6.5(e), apresentam os resultados de previsão de precipitação obtidos por diferentes redes neurais:

abordagem empírica (NN:Empírica), usando técnicas de otimização mono-objetiva (NN:MPCA) e multi-objetivo (NN:NSGA-II) e abordagem de redução de dados (TCA:MPCA), respectivamente. As Figuras 6.6(f), 6.6(g), 6.6(h) e 6.6(i), mostram os mapas de erros para os quatro modelos de previsão.

Assim como os resultados obtidos em outras estações (Verão, Outono, Inverno), na previsão gerada para a estação Primavera, constata-se que os padrões de precipitação obtidos usando a rede TCA:MPCA permanecem semelhantes ao campo de observação – ver Figura 6.6(e).

Desta forma, verifica-se a vantagem de utilizar uma abordagem de redução de dados na busca de extrair informações relevantes presentes nos dados, visto que, houve uma melhora significativa no processo de previsão de precipitação para a região Nordeste.



(a) Campo de precipitação de reanálise do NCEP/NCAR: Verão no Nordeste

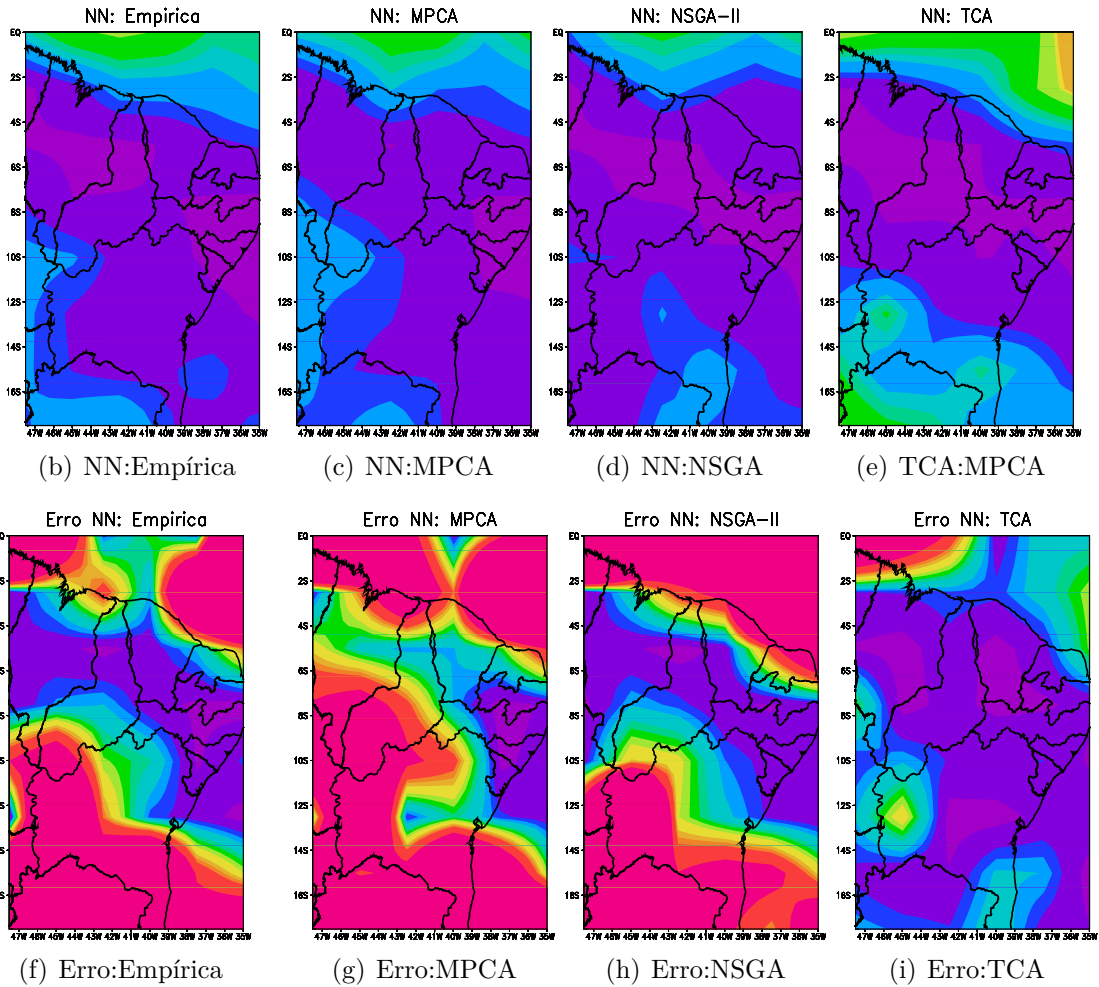
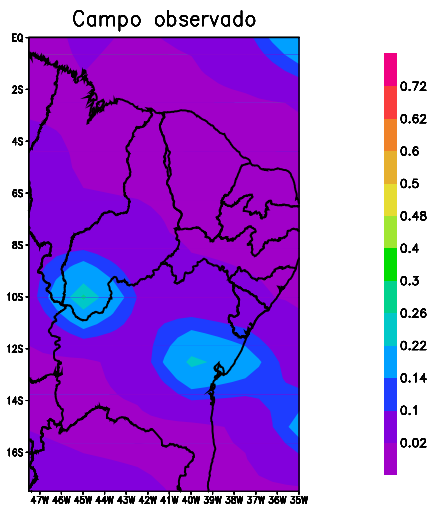


Figura 6.3 - Previsão sazonal de precipitação para a estação Verão



(a) Campo de precipitação de reanálise do NCEP/NCAR: Outono no Nordeste

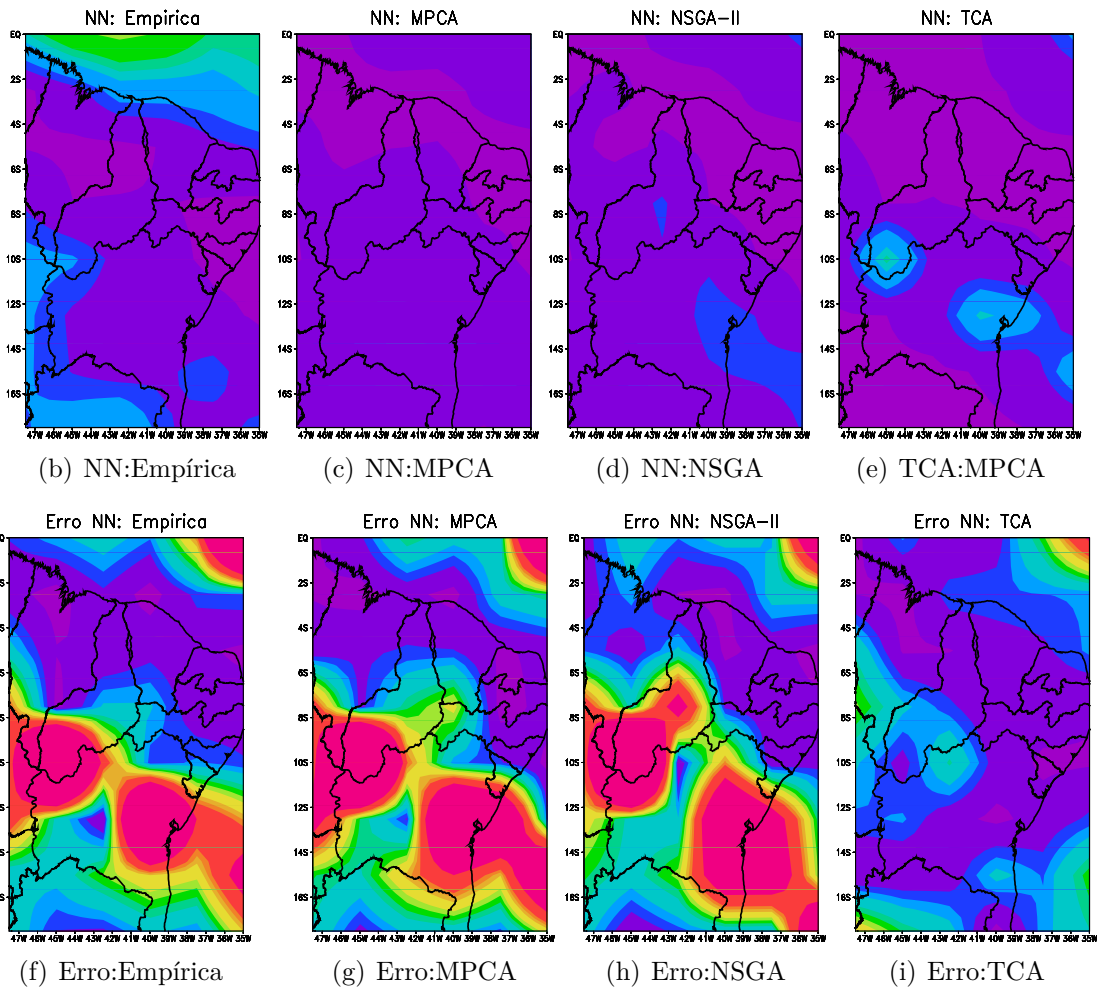
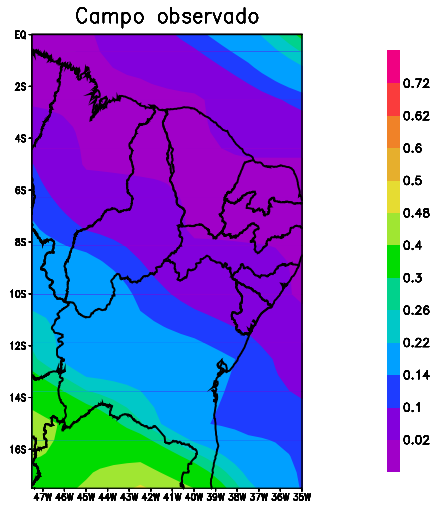


Figura 6.4 - Previsão sazonal de precipitação para a estação Outono



(a) Campo de precipitação de reanálise do NCEP/NCAR: Inverno no Nordeste

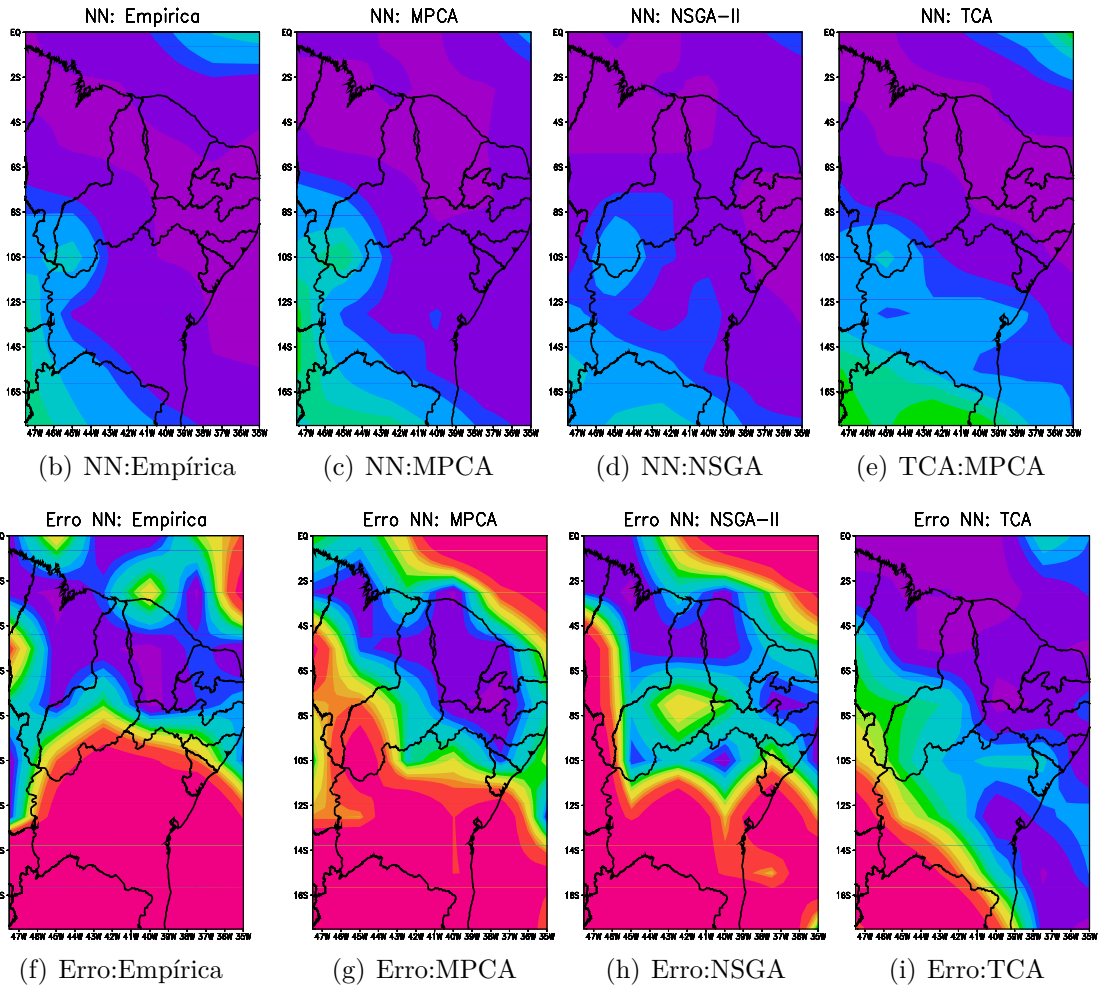
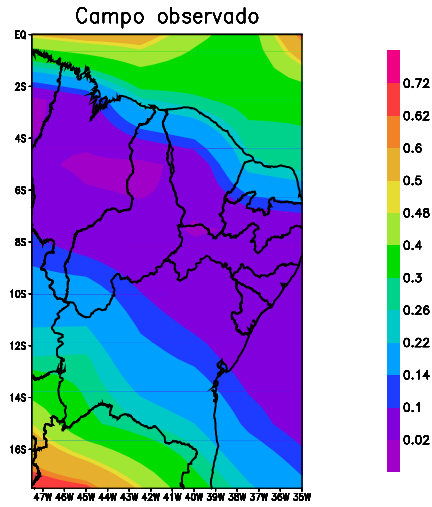


Figura 6.5 - Previsão sazonal de precipitação para a estação Inverno



(a) Campo de precipitação de reanálise do NCEP/NCAR: Primavera no Nordeste

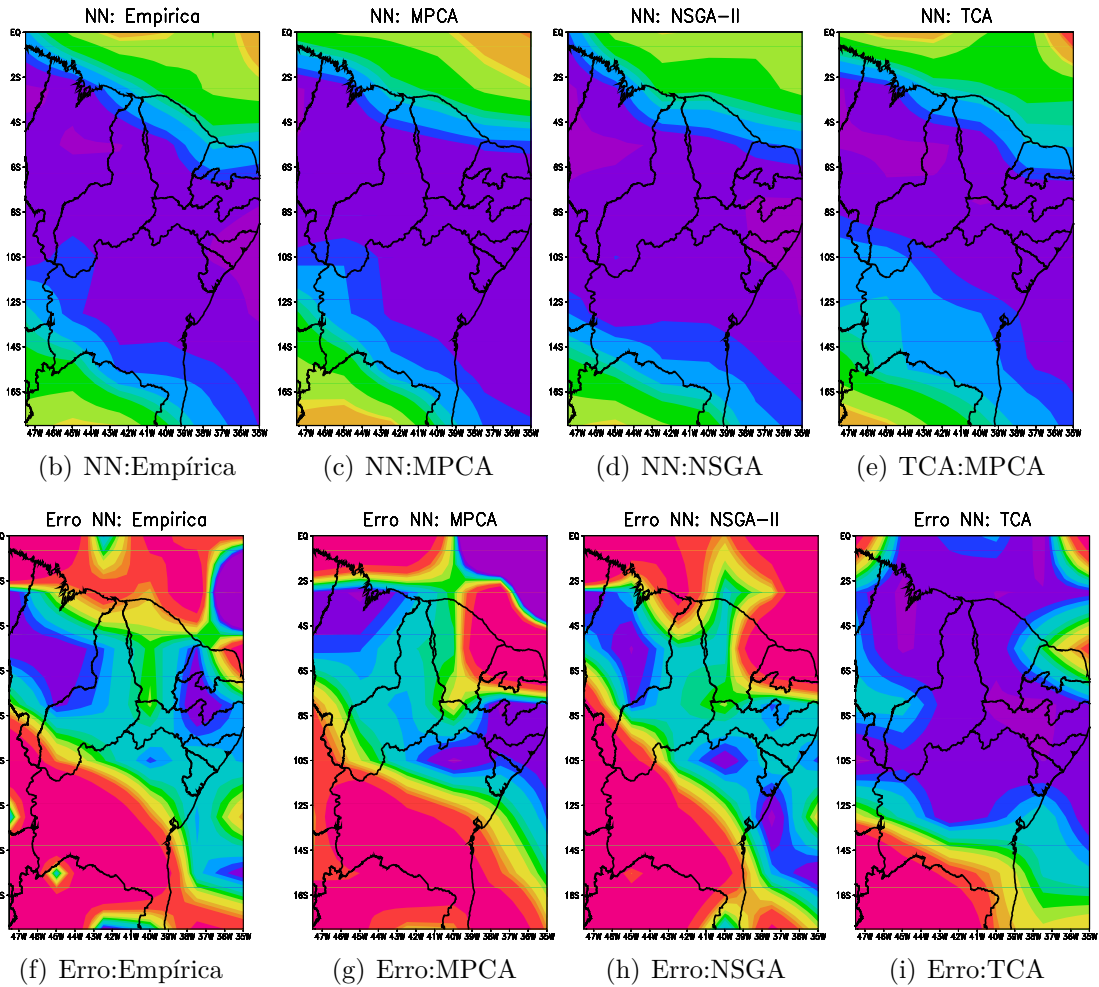


Figura 6.6 - Previsão sazonal de precipitação para a estação Primavera

6.3.1.2 Região Sul

Para a realização de previsão de precipitação na escala sazonal, parte da região Sul do Brasil foi selecionada. As coordenadas geográficas que delimitam a região considerada estão compreendidas entre as longitudes 62°W, 46°W e latitudes 35°S, 25°S, como ilustrado na [Figura 6.7](#) (quadrado vermelho no mapa do Brasil).

Alguns fenômenos atmosféricos que atuam sobre a região Sul do Brasil são essenciais na determinação da climatologia de temperatura e precipitação. Os sistemas frontais sobre a região são os mais frequentes causadores de precipitação durante o ano todo (OLIVEIRA, 1986).

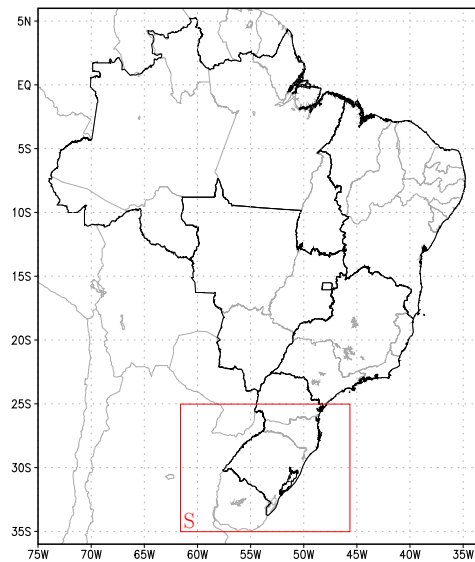


Figura 6.7 - Área de estudo: Sul (S)

Na [Tabela 6.6](#) são apresentados os resultados médios obtidos em 15 experimentos. Para efeito de comparação, foram avaliados o erro quadrático médio (EQM) do conjunto de teste, o número médio dos parâmetros, o melhor e o pior conjunto de parâmetros da rede neural ao final da otimização. Verifica-se que os experimentos obtidos com a meta-heurística NSGA-II, obteve uma menor complexidade em relação às abordagens NN-Empírica e NN-MPCA.

Redução de variáveis meteorológicas

As variáveis mais significativas obtidas pela TCA para a região Sul são apresentadas na [Tabela 6.7](#). Verifica-se que do total de variáveis apresentadas na [Tabela 6.1](#), sete

Tabela 6.6 - Resultados para a previsão climática na região Sul

Região	Método	Estatística	$n_{\text{neurônios}}$	$f_{\text{ativação}}$	η	α	EQM
Sul	Empírica	–	18	1	0.4	0.9	1.47E-03
	MPCA	Melhor	13	1	0.1387	0.9968	3.00E-04
		Pior	19	1	0.7159	0.9836	1.24E-03
		Média	26	1	0.4695	0.9638	4.29E-04
	NSGA-II	Melhor	2	2	0.5731	0.2396	2.00E-04
		Pior	2	1	0.9994	0.4484	3.76E-03
Média		1	2	0.3515	0.0243	2.70E-03	

foram extraídas para o processo de previsão de precipitação (ANOCHI, 2010).

Tabela 6.7 - Variáveis extraídas pela TCA - Sul

Variável	% (f_A)
temp	77
u300	79
u850	80
v300	77
v500	79
v850	71
shum	78

Precipitação: Verão de 2014

A Figura 6.8(a) mostra o campo de precipitação de reanálise para a estação Verão de 2014 na região Sul do Brasil. As Figuras 6.8(b), 6.8(c), 6.8(d) e 6.8(e) são os resultados de previsão de precipitação obtidos com diferentes NNs: NN:Empírica – configurada por experimentação empírica, NN:MPCA – configurada usando a meta-heurística MPCA, NN:NSGA – configurada usando algoritmo multiobjetivo NSGA-II e TCA:MPCA – treinada com conjunto reduzido de variáveis e configurada com os parâmetros ótimos obtidos com o uso do MPCA, respectivamente. As Figuras 6.9(a), 6.9(b), 6.9(c) e 6.9(d) mostram os mapas de erro obtidos a partir dos modelos: NN:Empírica, NN:MPCA, NN:NSGA e TCA:MPCA, respectivamente.

Analisando os resultados apresentados na Figura 6.8, verifica-se que todos os modelos de redes neurais (NN:Empírica, NN:MPCA, NN:NSGA, e TCA:MPCA) apresentaram previsões bem próxima do observado, especialmente, no estado do Rio Grande do Sul, seguindo com uma baixa margem de erro. Por outro lado, as me-

lhores respostas são os modelos NN:NSGA e TCA:MPCA, as quais identificaram o evento de precipitação nos estados do Rio Grande do Sul, Santa Catarina e Paraná. Entretanto, a previsão obtida usando a rede TCA:MPCA, detectou os níveis de precipitação com uma precisão significativa – ver 6.9(d).

Precipitação: Outono de 2014

A Figura 6.10(a) exhibe o campo de precipitação de reanálise para a estação Outono em 2014. As Figuras 6.10(b), 6.10(c), 6.10(d) e 6.10(e) apresentam os resultados de previsão de precipitação obtidos com as quatro diferentes redes: NN:Empírica, NN:MPCA, NN:NSGA e TCA:MPCA, respectivamente. As Figuras 6.11(a), 6.11(b), 6.11(c) e 6.11(d) mostram os mapas de erro adquiridos a partir das respostas dos modelos: NN:Empírica, NN:MPCA, NN:NSGA e NN:TCA, respectivamente.

Analisando os resultados obtidos para a estação Outono, constata-se que as previsões obtidas com os modelos NN:NSGA e TCA:MPCA, identificaram os níveis de precipitação presentes no campo de observação (ver Figura 6.10(a)), seguido com uma margem de erro concentrada no intervalo $[0, 0,005]$ enquanto que o erro da NN-Empírica encontra-se no intervalo $[0,005; 0,9]$ e para a NN-MPCA no intervalo $[0,5; 0,9]$ – ver Figuras 6.11(c) e 6.11(d).

Precipitação: Inverno de 2014

A Figura 6.12(a) mostra o campo de precipitação para a estação Inverno de 2014. As Figuras 6.12(b), 6.12(c) e 6.12(d) são os resultados de previsão de precipitação obtidos com NN-Empírica, NN-MPCA e NN:NSGA, respectivamente. As Figuras 6.13(a), 6.13(b) e 6.13(c) apresentam os mapas de erro para a previsão de precipitação obtidos por: NN:Empírica, NN:MPCA, NN:NSGA e NN:TCA, respectivamente.

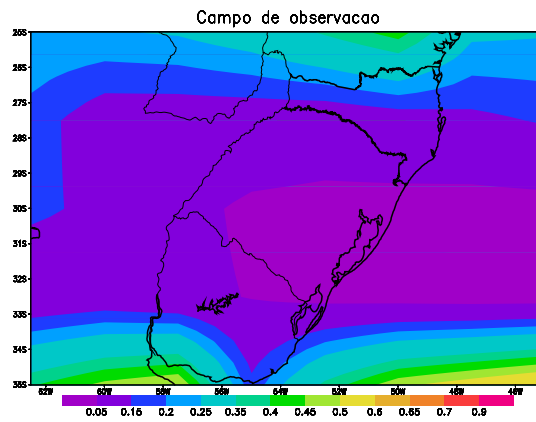
Tendo em vista os resultados mostrados na Figura 6.12, fica evidente que a melhor solução é a rede TCA:MPCA, a qual utilizou as variáveis selecionadas pela TCA e os parâmetros ótimos obtido com o MPCA, seguido com uma margem de erro no centrada no intervalo $[0,05; 0,5]$ – ver Figura 6.13(d). Embora a NN:NSGA siga um padrão de precipitação muito semelhante ao observado, possui um erro superior, assim como as redes NN:MPCA e NN:Empírica.

Precipitação: Primavera de 2014

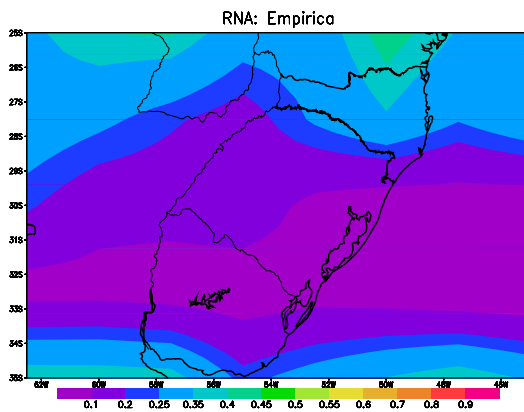
Na Figura 6.14(a) é apresentado o campo de precipitação de reanálise para a estação Primavera em 2014. As Figuras 6.14(b), 6.14(c), 6.14(d) e 6.14(e) exibem os resul-

tados de previsão de precipitação obtidos com as redes: NN:Empírica, NN:MPCA, NN:NSGA e TCA:MPCA, respectivamente. As Figuras 6.15(a), 6.15(b), 6.15(c) e 6.15(d) mostram os mapas de erro obtidos por: NN:Empírica, NN:MPCA, NN:NSGA e NN:TCA, respectivamente.

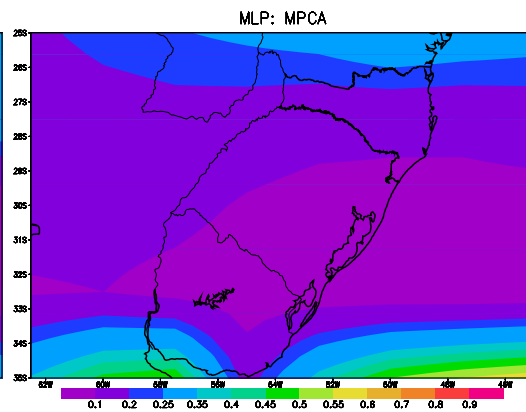
Levando-se em consideração os resultados apresentados na Figura 6.14, verifica-se, novamente que a previsão obtida com a rede TCA:MPCA foi a melhor solução obtida para o processo de previsão para a estação Primavera, seguido com uma baixa margem de erro. Ainda que, a rede NN:NSGA tenha identificado padrões de precipitação semelhantes ao campo de observação (Figura 6.14(a)) nos estados de Santa Catarina e Paraná, a solução usando abordagem de redução de variáveis TCA:MPCA, foi melhor e com erro inferior em relação aos outros modelos de redes NN:Empírica, NN:MPCA e NN:NSGA – ver Figura 6.15(d).



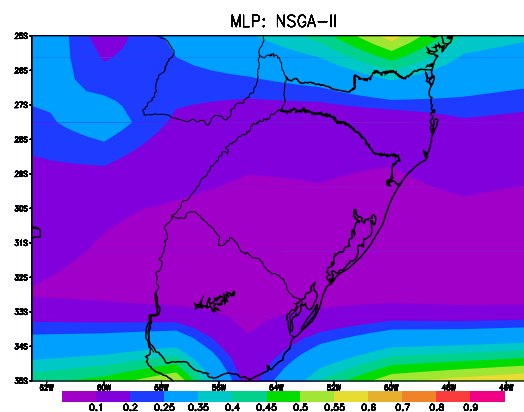
(a) Campo de precipitação de reanálise do NCEP/NCAR: Verão no Sul



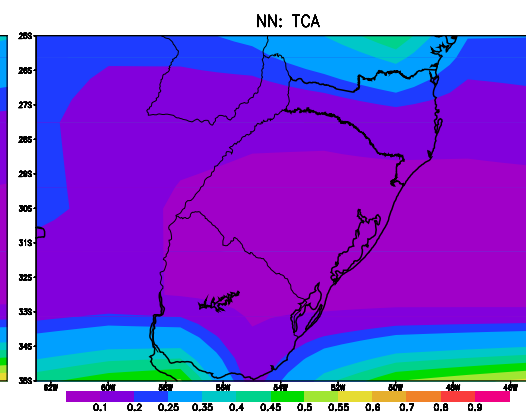
(b) NN:Empírica



(c) NN:MPCA



(d) NN:NSGA



(e) TCA:MPCA

Figura 6.8 - Previsão sazonal de precipitação para a estação Verão

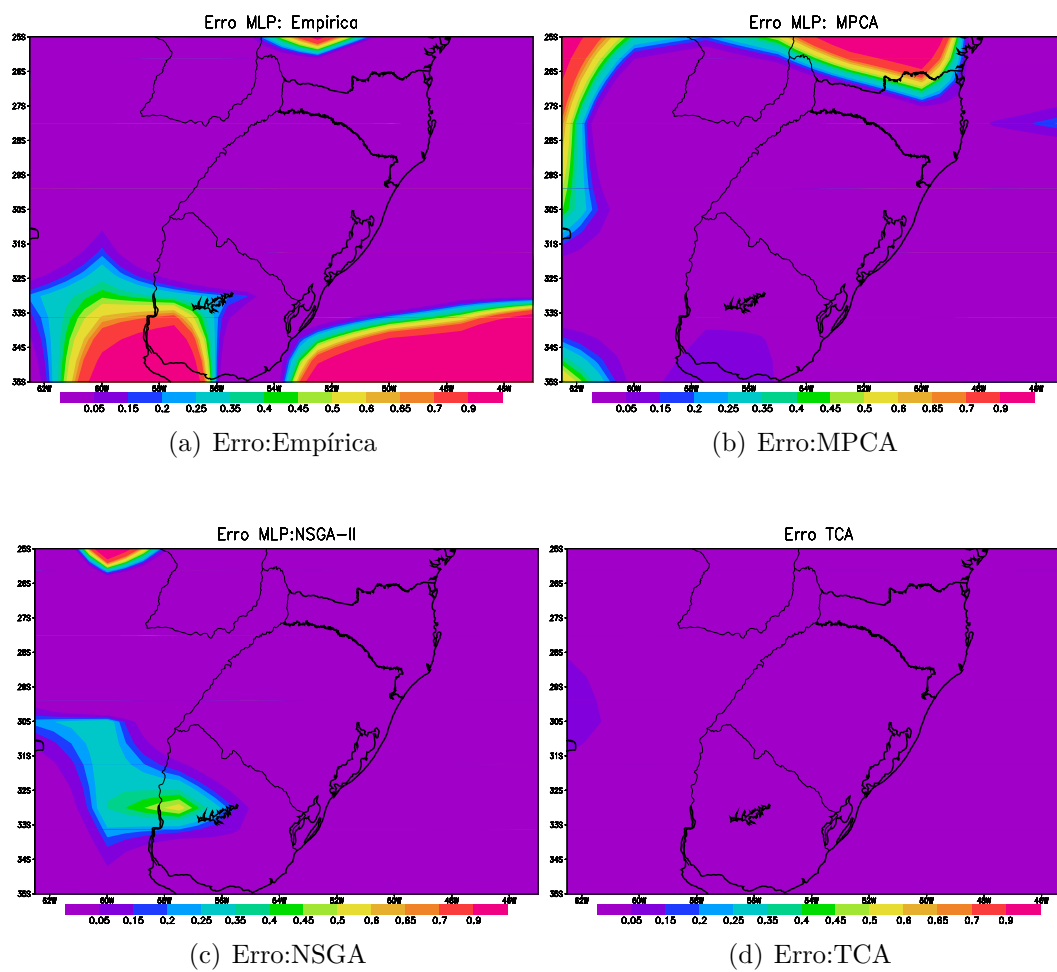
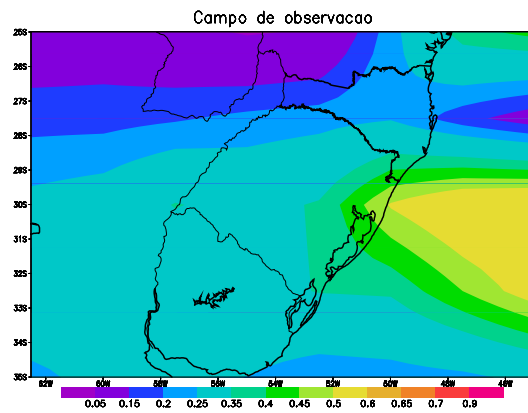
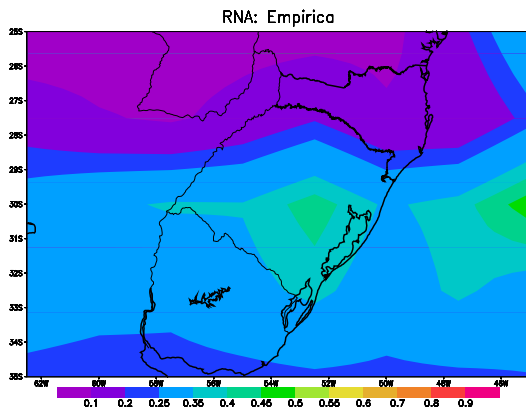


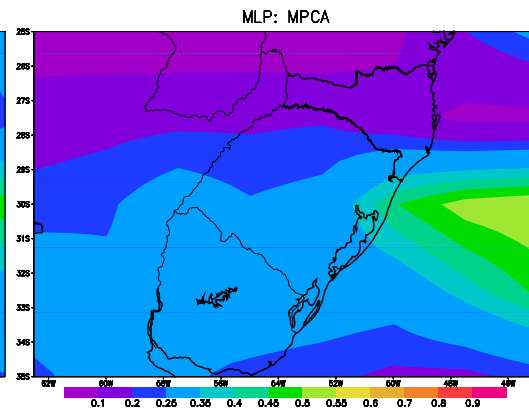
Figura 6.9 - Mapa de erro para previsão sazonal de precipitação na estação Verão



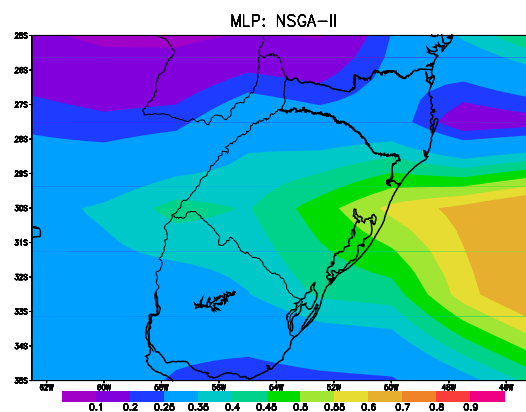
(a) Campo de precipitação de reanálise do NCEP/NCAR: Outono no Sul



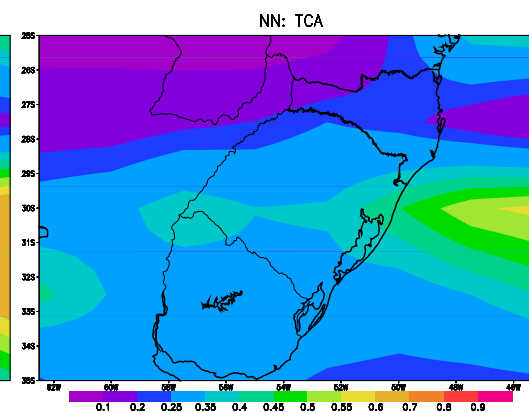
(b) NN:Empírica



(c) NN:MPCA



(d) NN:NSGA



(e) TCA:MPCA

Figura 6.10 - Previsão sazonal de precipitação para a estação Outono

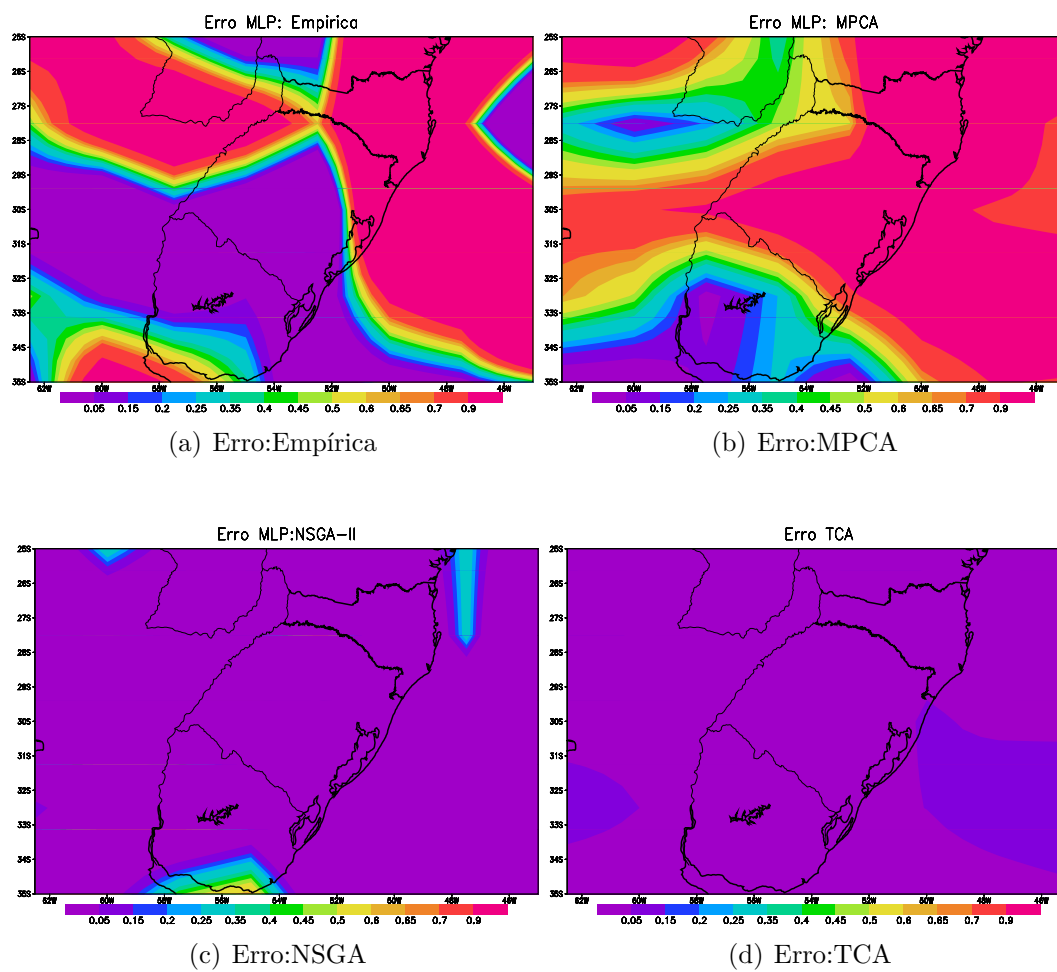
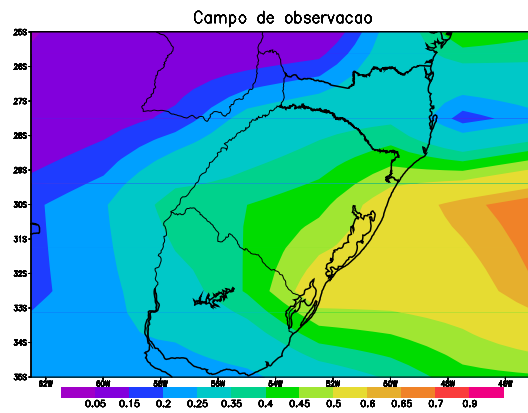
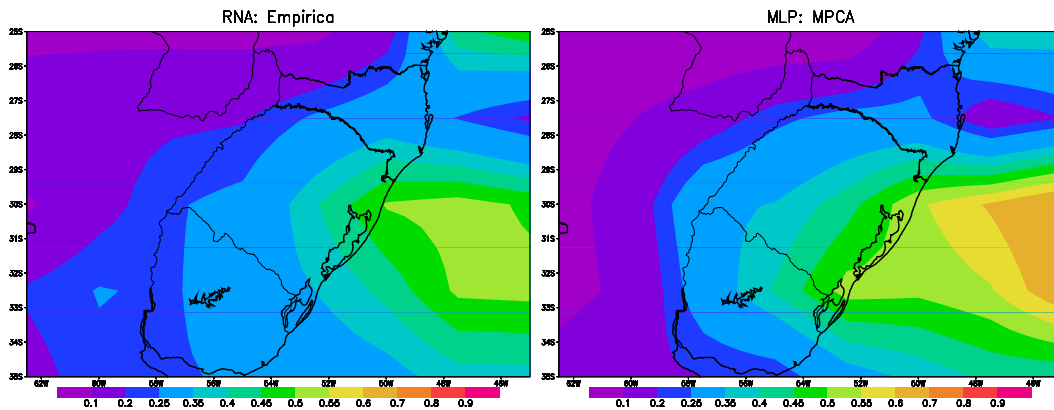


Figura 6.11 - Mapa de erro para previsão sazonal de precipitação na estação Outono

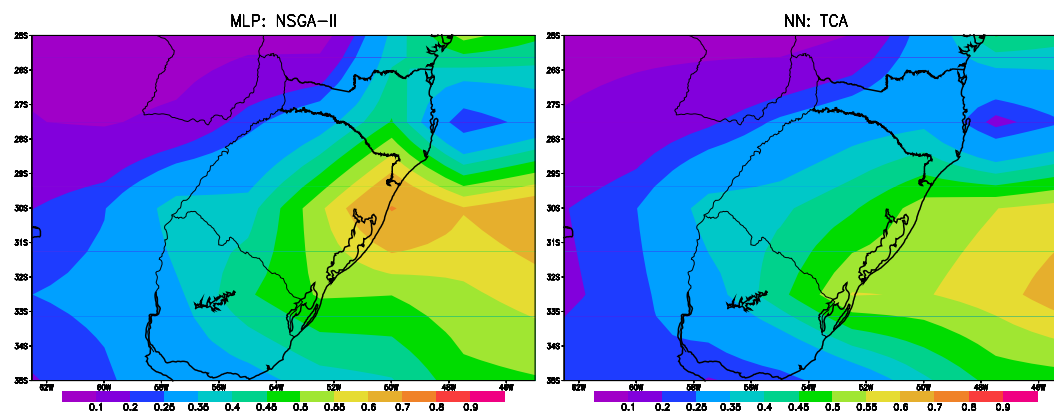


(a) Campo de precipitação de reanálise do NCEP/NCAR: Inverno no Sul



(b) NN:Empírica

(c) NN:MPCA



(d) NN:NSGA

(e) TCA:MPCA

Figura 6.12 - Resultados de previsão sazonal de precipitação na estação Inverno

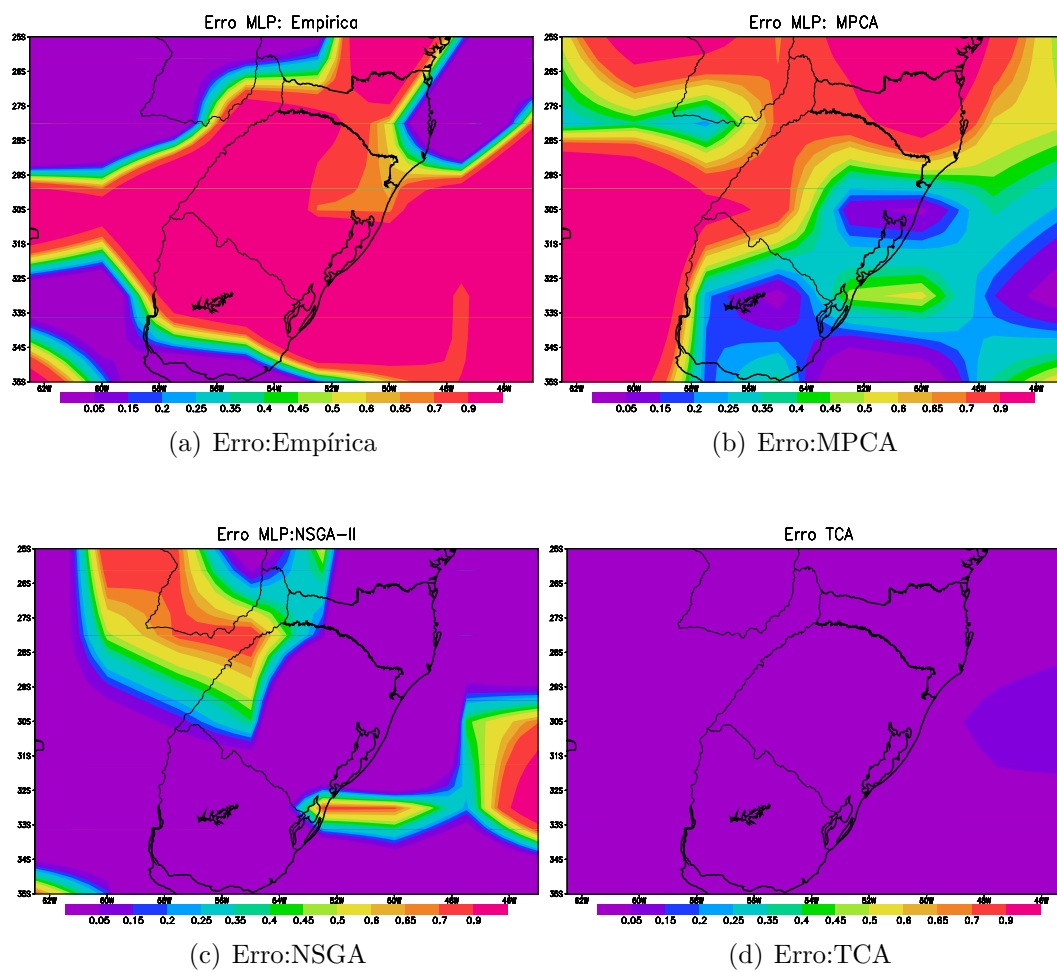
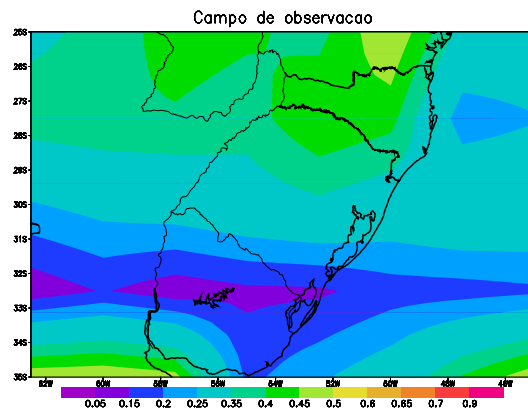
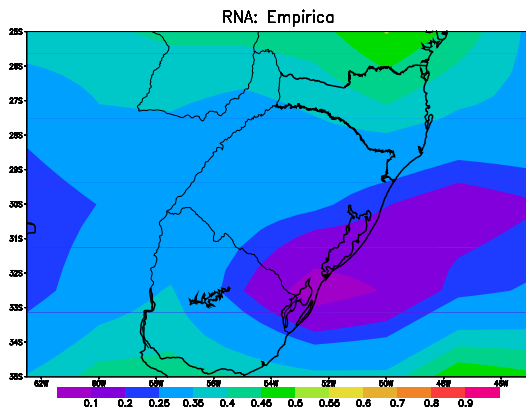


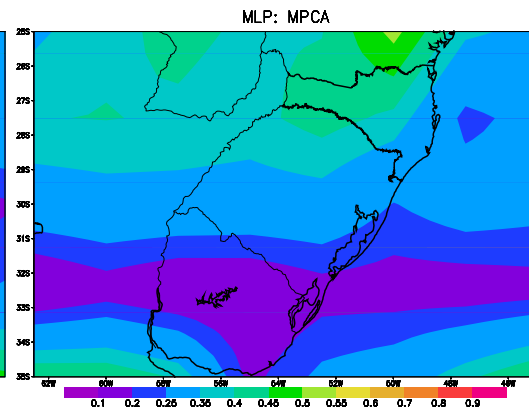
Figura 6.13 - Mapa de erro para previsão sazonal de precipitação na estação Inverno



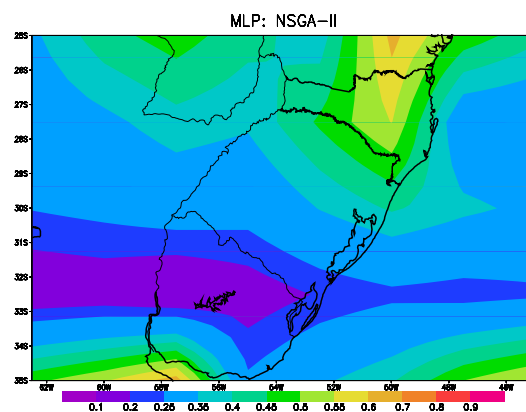
(a) Campo de precipitação de reanálise do NCEP/NCAR: Primavera no Sul



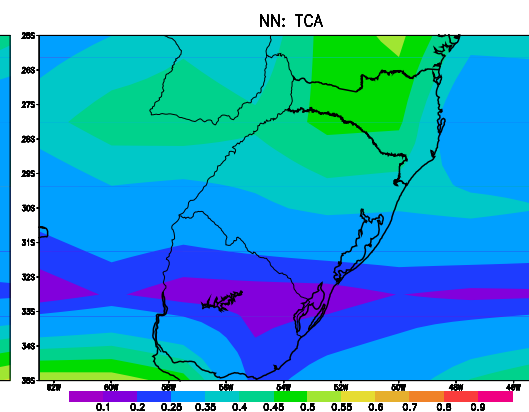
(b) NN:Empírica



(c) NN:MPCA



(d) NN:NSGA



(e) TCA:MPCA

Figura 6.14 - Resultados de previsão sazonal de precipitação para a estação Primavera

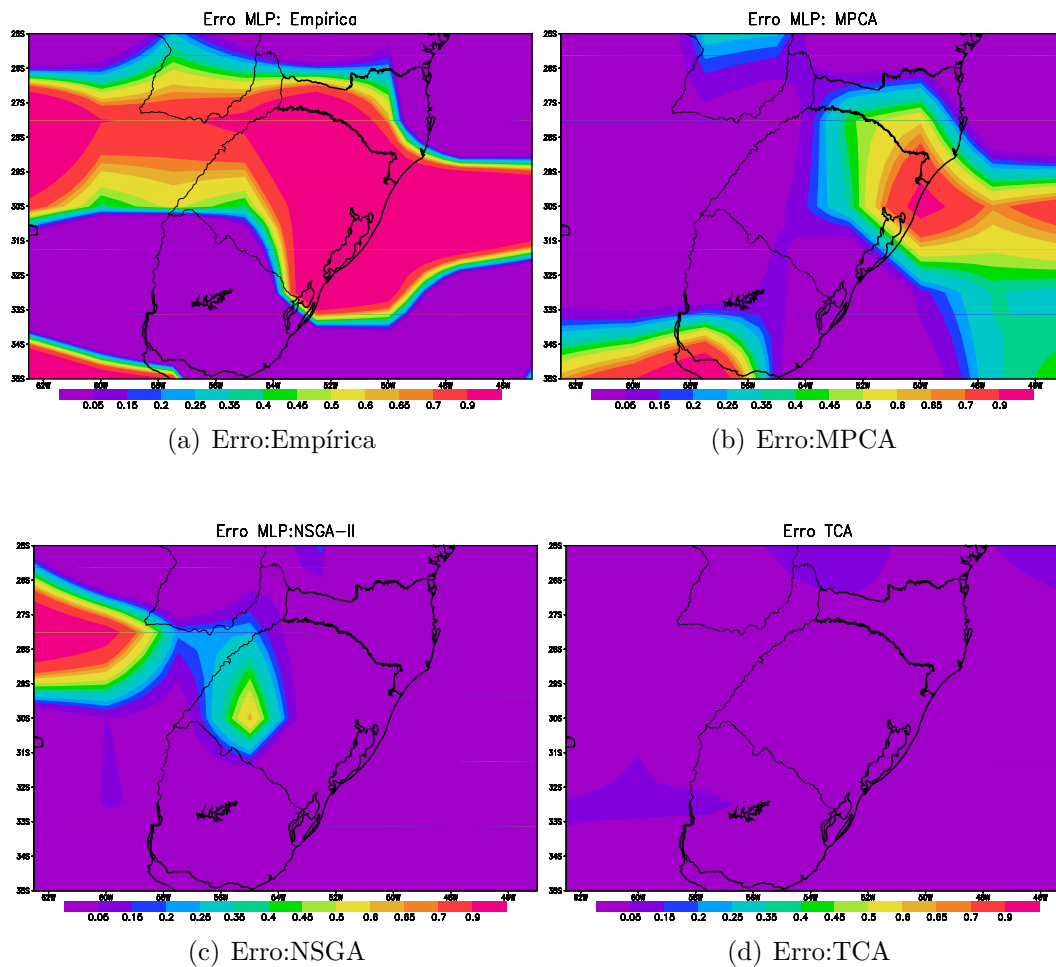


Figura 6.15 - Mapa de erro para previsão sazonal de precipitação para a estação Primavera

6.3.1.3 Região Sudeste

A região Sudeste do Brasil foi selecionada para realizar a previsão climática de precipitação na escala sazonal. As coordenadas geográficas que delimitam a região considerada estão compreendidas entre as longitudes 52°W, 38°W e latitudes 25°S, 15°S, como ilustrado na [Figura 6.16](#) (quadrado vermelho no mapa do Brasil).

A região Sudeste do Brasil é caracterizada pela atuação de sistemas meteorológicos que associam propriedades de sistemas tropicais com sistemas típicos de latitudes médias. Durante os meses de maior atividade convectiva, a ZCAS é um dos principais fenômenos que influenciam o regime de precipitação ([QUADRO et al., 1996](#)).

Em geral a precipitação distribui-se uniformemente no Sudeste. Entretanto, iniciou-se em fevereiro de 2014 uma crise hídrica em todo o Sudeste. No estado de São Paulo, o nível do principal reservatório do estado, o Sistema Cantareira, que abastece a capital paulista, registrou os menores valores de sua história.

A região Sudeste apresenta baixa previsibilidade. Assim, uma previsão sobre o início de uma estação chuvosa e/ou seca, é um alvo difícil de atingir. O CPTEC e conceituados centros meteorológicos dos Estados Unidos (NCAR, NCEP, *Center for Ocean, Land and Atmosphere Studies* (COLA)), dentre outros, reconhecem a complexidade em prever o clima sazonal na região Sudeste do Brasil. As previsões apresentam um baixo grau de confiabilidade, especialmente para chuvas durante estações intermediárias como primavera e outono ([CPTEC, 2015a](#)).

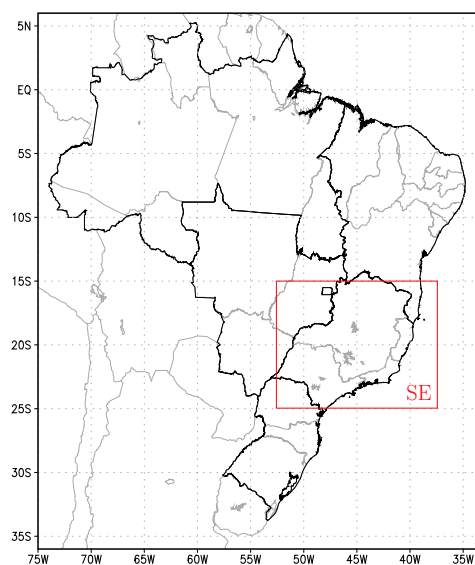


Figura 6.16 - Área de estudo: Sudeste (SE)

Na [Tabela 6.8](#) são apresentados os resultados médios obtidos em 15 experimentos. Para efeito de comparação, foram avaliados o erro quadrático médio (EQM) do conjunto de teste, o número médio dos parâmetros, o melhor e o pior conjunto de parâmetros da rede neural ao final da otimização.

Tabela 6.8 - Resultados para a previsão climática na região Sudeste

Região	Método	Estatística	$n_{\text{neurônios}}$	$f_{\text{ativação}}$	η	α	EQM	
Sudeste	Empírica	–	15	1	0.4	0.9	2.32E-02	
	MPCA	Melhor	8	1	0.1718	0.2201	3.01E-02	
		Pior	5	2	0.8918	0.0309	3.09E-02	
		Média	6	1	0.5635	0.3457	3.07E-02	
	NSGA-II	Melhor		5	2	0.7016	0.9810	2.05E-02
		Pior		5	2	0.5390	0.7462	2.22E-02
		Média		5	2	0.6163	0.5977	2.20E-02

Redução de variáveis meteorológicas

A [Tabela 6.9](#) apresenta os atributos mais relevantes para o processo de previsão na região Sudeste. Observa-se que dentre as dez variáveis meteorológicas disponíveis na base de dados ([Tabela 6.1](#)), foram selecionadas cinco variáveis para o processo de previsão climática de precipitação na região Sudeste do Brasil ([ANOCHI, 2010](#)).

Tabela 6.9 - Variáveis extraídas pela TCA - Sudeste

Variável	% (f_A)
temp	78
u850	79
v300	74
v500	80
shum	89

Precipitação: Verão de 2014

A [Figura 6.17\(a\)](#) mostra o campo de precipitação de reanálise para a estação Verão de 2014 na região Sudeste do Brasil. As [Figuras 6.17\(b\)](#), [6.17\(c\)](#), [6.17\(d\)](#) e [6.17\(e\)](#) são os resultados de previsão de precipitação obtidos com as redes: NN:Empírica – configuração por experimentação empírica, NN:MPCA – a topologia foi configurada automaticamente pela meta-heurística MPCA, NN:NSGA – configurada pela meta-heurística NSGA-II, e TCA:MPCA – configurada com os parâmetros ótimos

obtidos pelo MPCA e treinada com o conjunto reduzido de variáveis apresentados na Tabela 6.9. As Figuras 6.18(a), 6.18(b), 6.18(c) e 6.18(d) mostram os mapas de erro para a previsão de precipitação obtidos por: NN:Empírica, NN:MPCA, NN:NSGA e TCA:MPCA, respectivamente.

Analisando os resultados de previsão mostrados na Figura 6.17, verifica-se que a previsão produzida pelo modelo NN:MPCA, identificou um padrão de precipitação semelhante ao campo de observação nos estados de Minas Gerais, Espírito Santo e Rio de Janeiro, entretanto no estado de São Paulo não houve uma melhora.

Precipitação: Outono de 2014

Na Figura 6.19(a) é mostrado o campo de precipitação de reanálise para a estação Outono. As Figuras 6.19(b), 6.19(c), 6.19(d) e 6.19(e) mostram os resultados de previsão climática obtidos com as redes: NN:Empírica, NN:MPCA, NN:NSGA e TCA:MPCA, respectivamente. As Figuras 6.20(a), 6.20(b), 6.20(c) e 6.20(d) apresentam os mapas de erro para todos os modelos de previsão.

Diante dos resultados apresentados na Figura 6.19, constata-se que as previsões obtidas com as redes TCA:MPCA e NN:NSGA identificaram padrões de precipitação semelhantes ao campo de observação. Entretanto, fazendo uma análise qualitativa, verifica-se que a saída produzida com o modelo TCA:MPCA, detectou os níveis de precipitação em toda a região Sudeste, especialmente, no estado de São Paulo, área em que os outros modelos não identificaram a ocorrência de precipitação.

Precipitação: Inverno de 2014

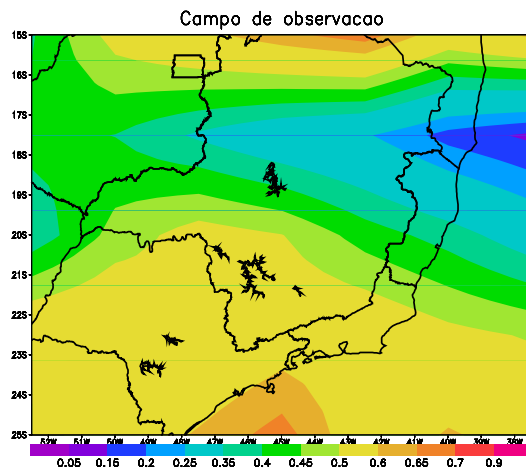
A 6.21(a) mostra o campo de precipitação de reanálise para a estação Inverno de 2014. As Figuras 6.21(b), 6.21(c), 6.21(d) e 6.21(e) apresentam os resultados de previsão de precipitação obtidos com os modelos: NN:Empírica, NN:MPCA, NN:NSGA e TCA:MPCA, respectivamente. As Figuras 6.22(a), 6.22(b), 6.22(c) e 6.22(d) apresentam os mapas de erro para todos os modelos de previsão.

Levando-se em consideração o resultado apresentado na Figura 6.21, constata-se que o modelo NN:MPCA, identificou padrões de precipitação semelhantes ao observado, em particular, no estado de São Paulo. Entretanto, a previsão obtida pela rede NN:NSGA, verifica-se que nos estados do Espírito Santo e Rio de Janeiro, apresentou uma boa previsibilidade, exceto nos estados de São Paulo e Minas Gerais.

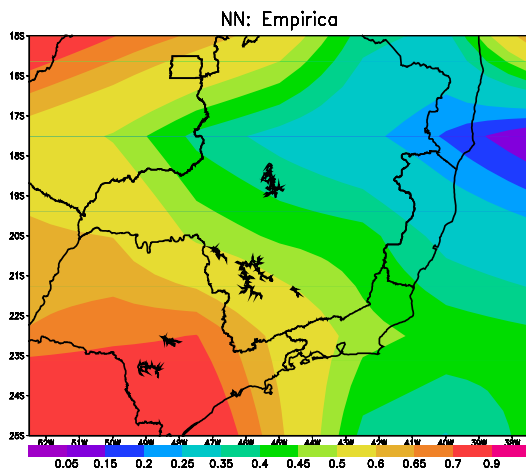
Precipitação: Primavera de 2014

A Figura 6.23(a) apresenta o campo de precipitação observado para a estação Primavera em 2014. As Figuras 6.23(b), 6.23(c), 6.23(d) e 6.23(e) exibem os resultados de previsão de precipitação obtidos com as redes: NN:Empírica, NN:MPCA, NN:NSGA e TCA:MPCA, respectivamente. As Figuras 6.24(a), 6.24(b), 6.24(c) e 6.24(d) apresentam os mapas de erro para as redes: NN:Empírica, NN:MPCA, NN:NSGA e TCA:MPCA, respectivamente.

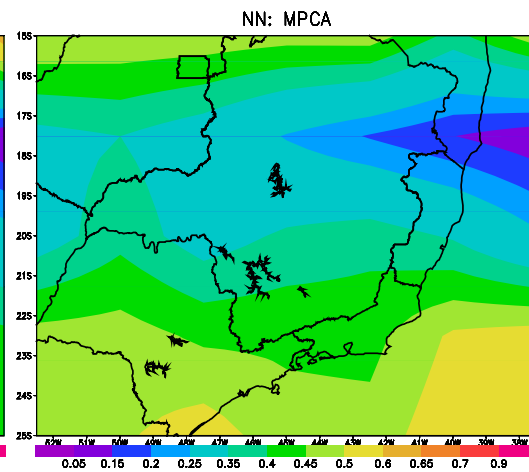
Tendo em vista os resultados apresentados na Figura 6.23, observa-se que a previsão obtida com a rede TCA:MPCA identificou níveis de precipitação nas regiões do Triângulo Mineiro e Sul de Minas no estado de Minas Gerais, seguido com uma baixa margem de erro. As previsões geradas com os modelos NN:Empírica, NN:MPCA e NN:NSGA tiveram baixa previsibilidade para a estação Primavera.



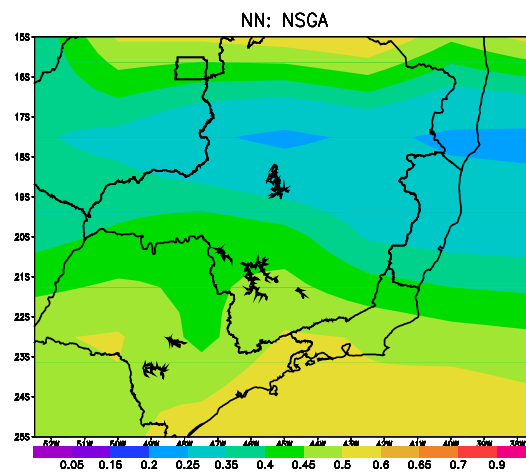
(a) Campo de precipitação de reanálise do NCEP/NCAR: Verão no Sudeste



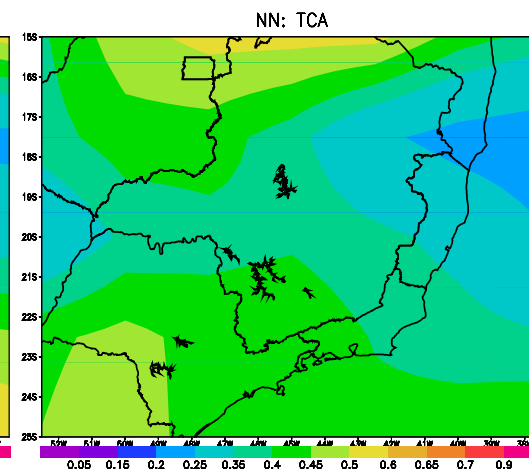
(b) NN:Empírica



(c) NN:MPCA



(d) NN:NSGA



(e) TCA:MPCA

Figura 6.17 - Resultados de previsão sazonal de precipitação para a estação Verão

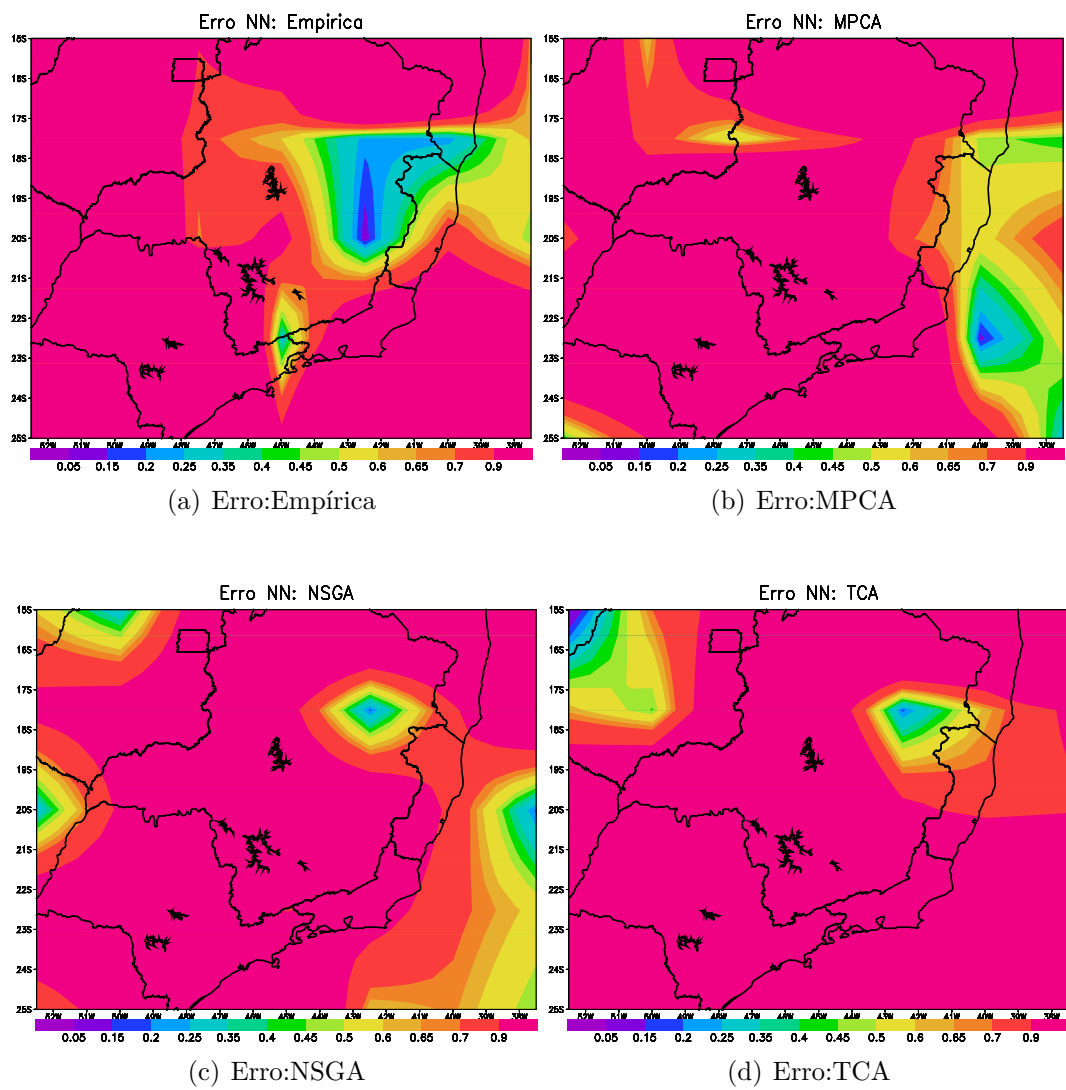
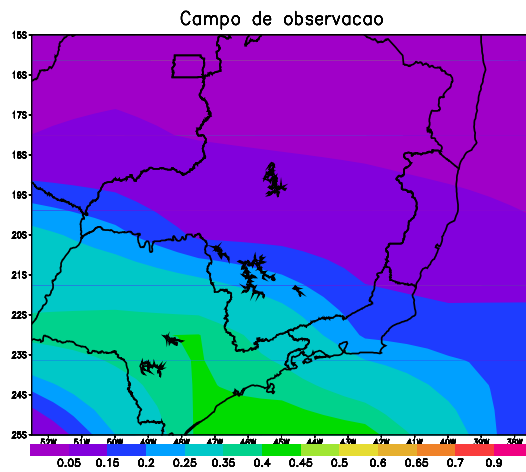
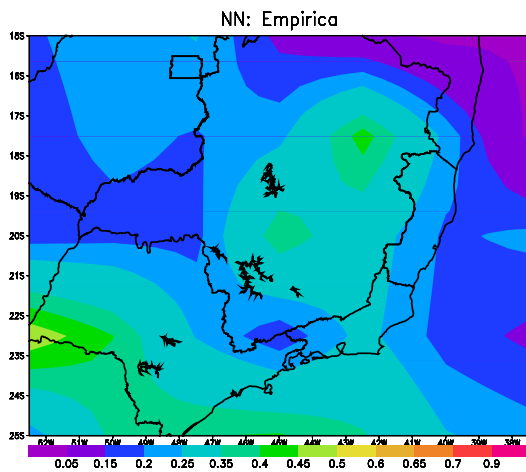


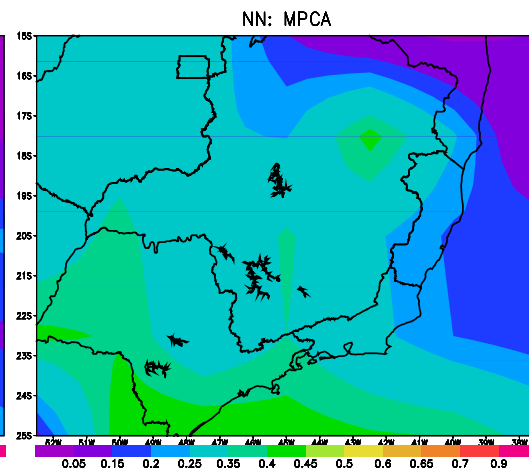
Figura 6.18 - Mapa de erro para a previsão sazonal de precipitação na estação Verão



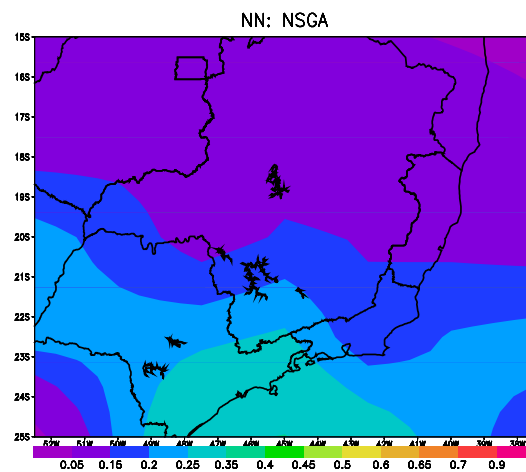
(a) Campo de precipitação de reanálise do NCEP/NCAR: Outono no Sudeste



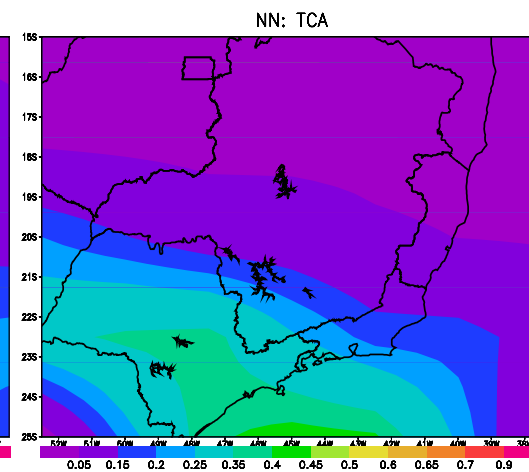
(b) NN:Empírica



(c) NN:MPCA



(d) NN:NSGA



(e) TCA:MPCA

Figura 6.19 - Resultados de previsão sazonal de precipitação para a estação Outono

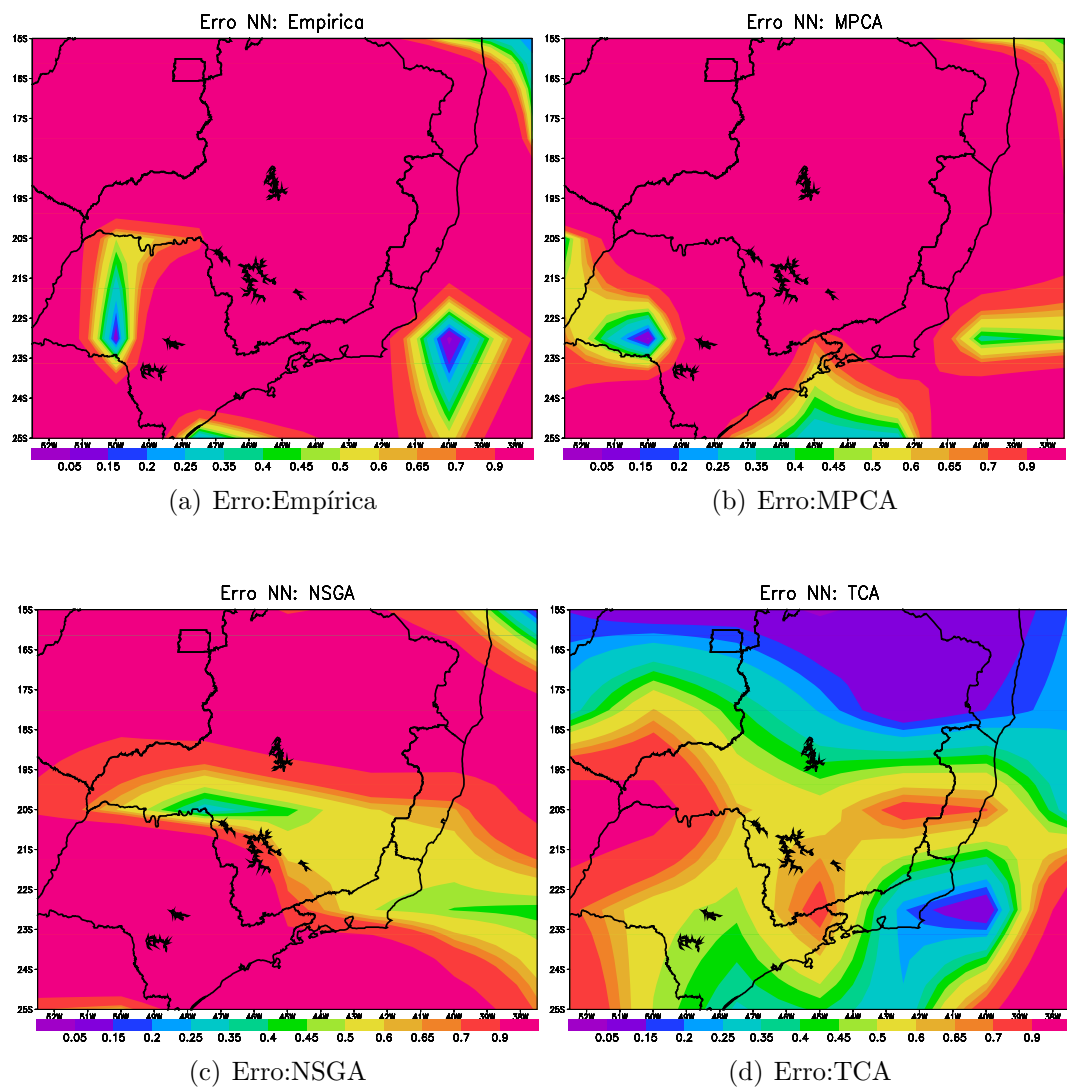
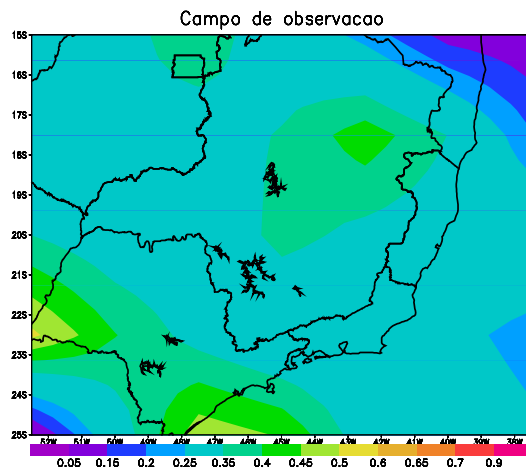
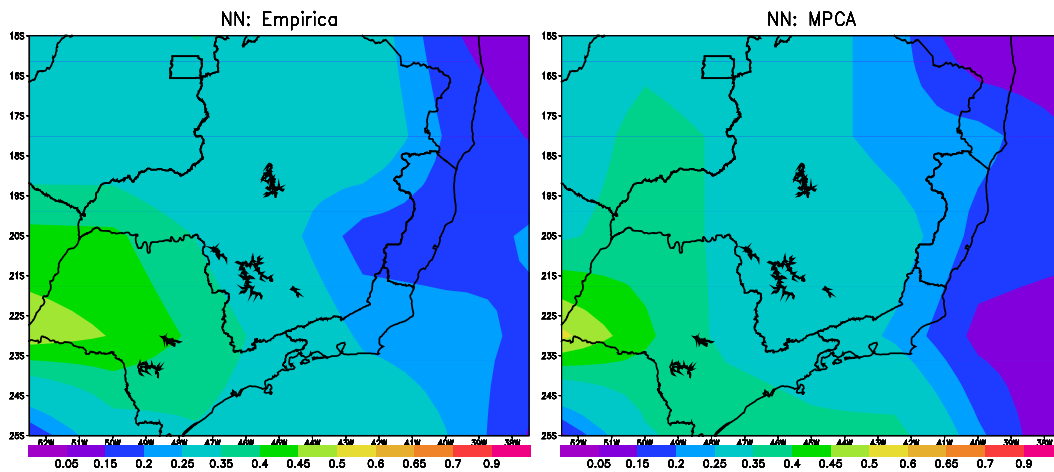


Figura 6.20 - Mapa de erro para a previsão sazonal de precipitação na estação Outono

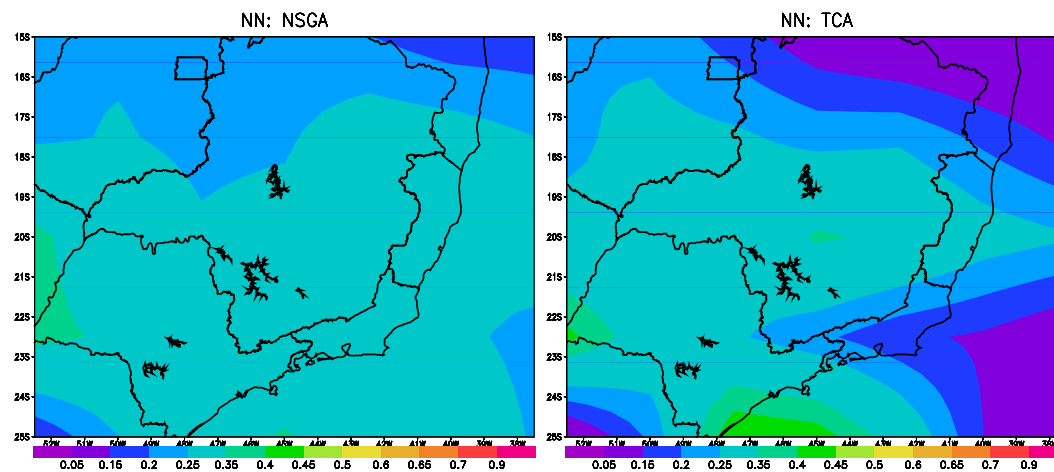


(a) Campo de precipitação de reanálise do NCEP/NCAR: Inverno no Sudeste



(b) NN:Empírica

(c) NN:MPCA



(d) NN:NSGA

(e) TCA:MPCA

Figura 6.21 - Resultados de previsão sazonal de precipitação para a estação Inverno

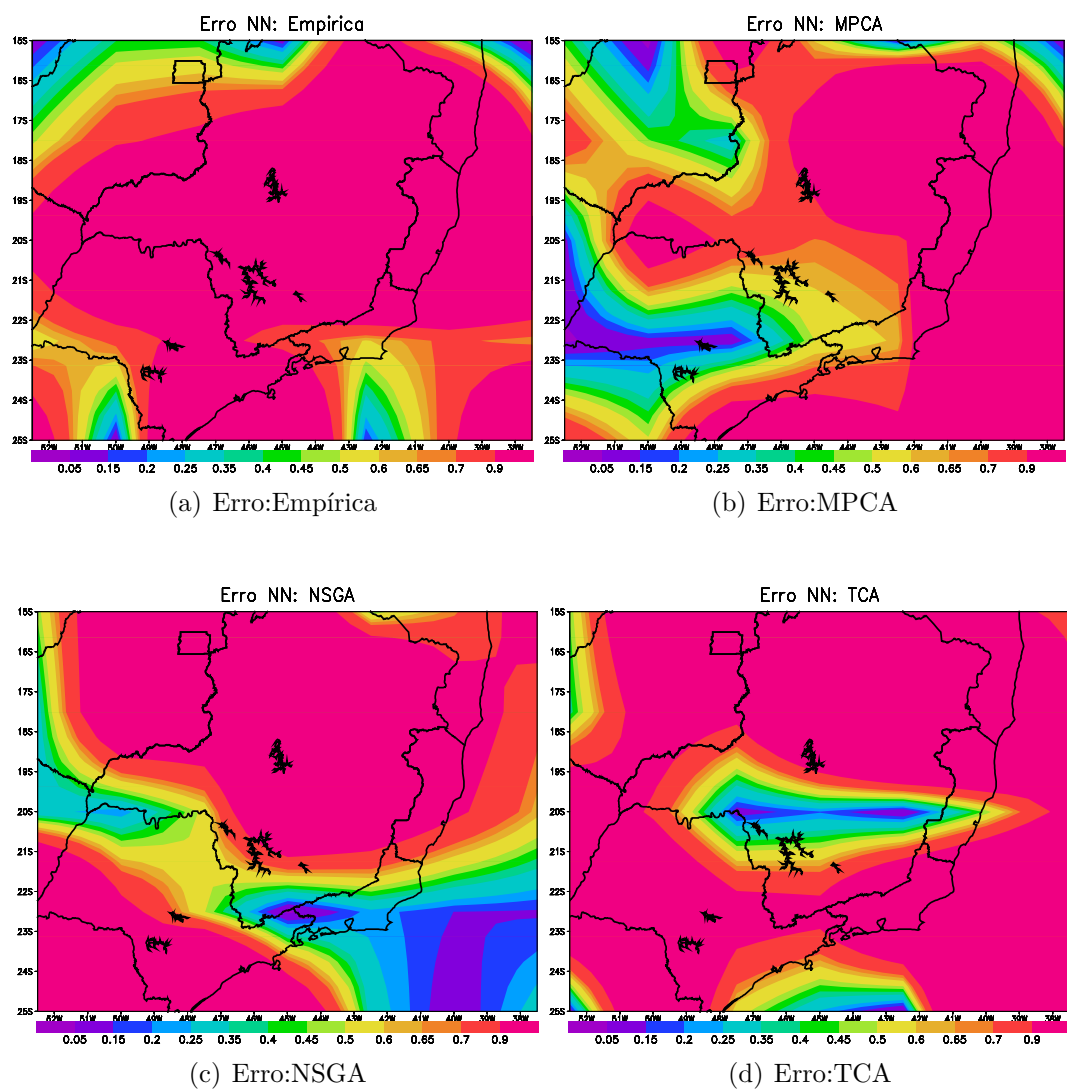
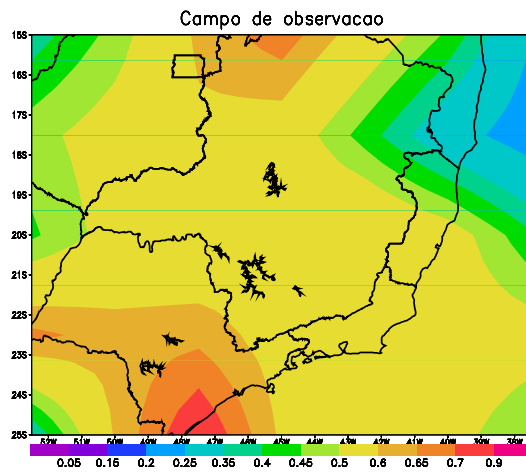
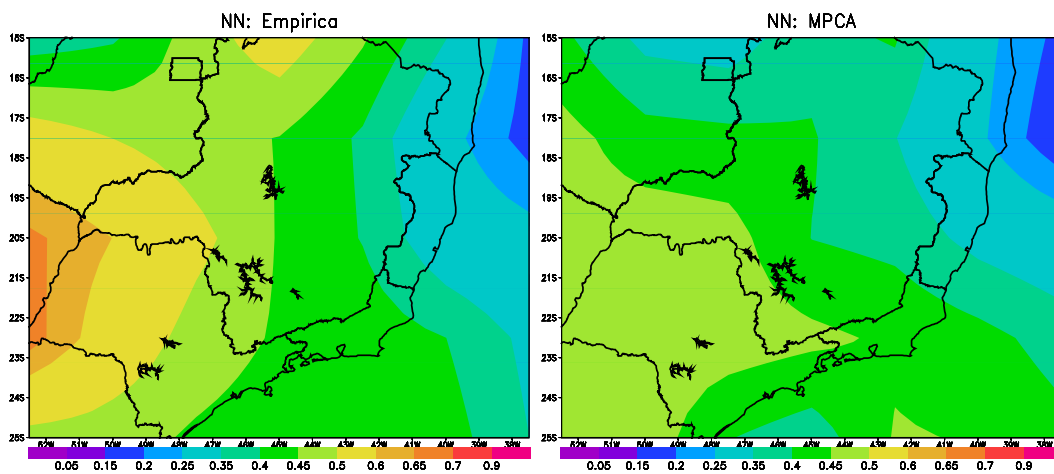


Figura 6.22 - Mapa de erro para a previsão sazonal de precipitação na estação Inverno

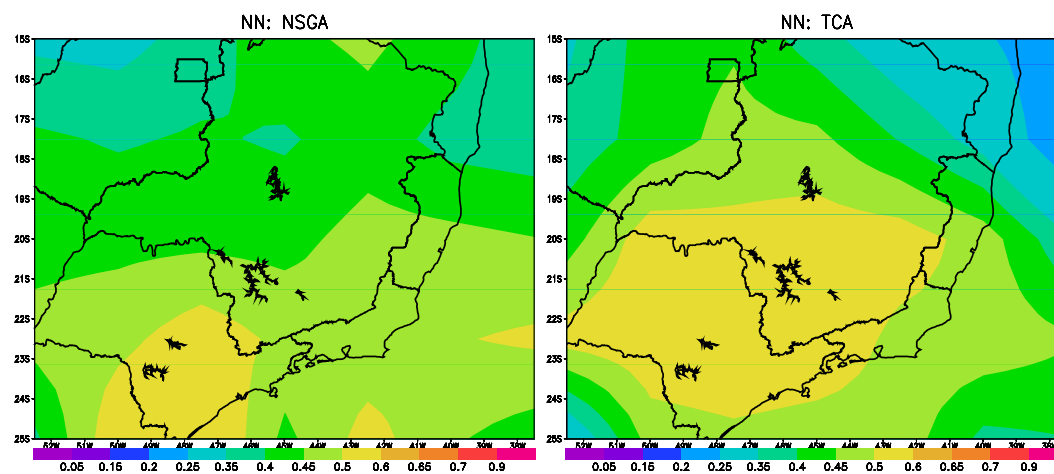


(a) Campo de precipitação de reanálise do NCEP/NCAR: Primavera no Sudeste



(b) NN:Empírica

(c) NN:MPCA



(d) NN:NSGA

(e) TCA:MPCA

Figura 6.23 - Resultados de previsão sazonal de precipitação para a estação Primavera

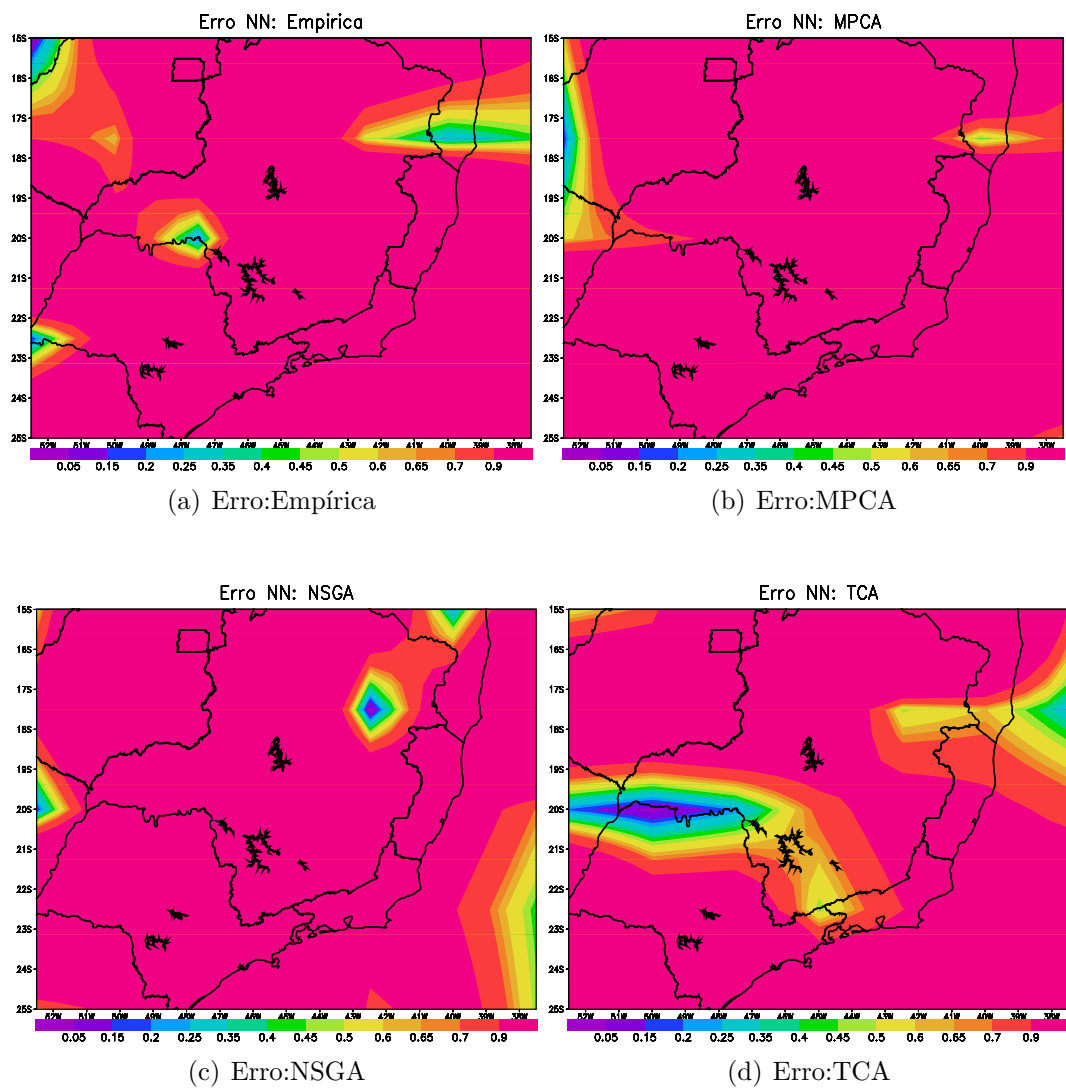


Figura 6.24 - Mapa de erro para a previsão sazonal de precipitação na estação Primavera

7 ESTRATÉGIA DE TREINAMENTO USANDO META-HEURÍSTICA

Neste capítulo, uma nova formulação para o treinamento de redes neurais supervisionadas é apresentada, em que a meta-heurística MPCA realiza o papel do algoritmo de treinamento *backpropagation* efetuando o ajuste dos principais parâmetros livres da rede e os pesos sinápticos do treinamento.

7.1 Método de treinamento usando MPCA

O processo de treinamento de redes neurais supervisionadas, consiste em encontrar os pesos sinápticos e níveis de *bias*, minimizando uma função custo, a qual expressa a diferença quadrática entre a saída desejada e a resposta produzida pela rede.

O ajuste dos pesos sinápticos é normalmente realizado através do algoritmo *backpropagation* (ou regra delta generalizada) (WILLIAMS; HINTON, 1986). Este é um método que utiliza o gradiente descendente do erro de classificação da rede como informação de retorno (*feedforward*) para a adaptação dos pesos. Originalmente, o algoritmo *backpropagation* obteve bom desempenho em problemas não-linearmente separáveis. Entretanto possui algumas limitações, tais como: convergência muito lenta; possibilidade de ficar preso em mínimos locais; alta complexidade computacional quando o número de parâmetros da rede é elevado e geralmente não há garantia que um bom treinamento seja alcançado (MARQUARDT, 1963).

Nesse trabalho, uma nova abordagem de treinamento é proposta para solucionar as limitações do algoritmo *backpropagation*, através da substituição do mesmo no ajuste dos pesos das conexões pela meta-heurística MPCA, a qual efetua o ajuste dos principais parâmetros livres (pesos sinápticos), minimizando a função objetivo, definida na Equação 7.1.

$$J(w_i) = \sum [\Phi_i^{\text{Obs}} - \Phi_i^{\text{RNA}}(w)], \quad (7.1)$$

em que Φ_i^{Obs} é a variável observada, Φ_i^{RNA} é a variável estimada pela rede neural e w é o vetor de pesos, sendo Φ o campo de precipitação.

7.1.1 Previsão mensal para a região Nordeste

Para a previsão de precipitação na escala mensal foram realizados estudos e experimentos sobre uma subárea da região Nordeste do Brasil. As coordenadas geográficas que delimitam a região considerada estão compreendidas entre as longitudes 47°W,

40°W e latitudes 10°S 0°, como ilustrado na Figura 7.1 (retângulo vermelho no mapa do Brasil).

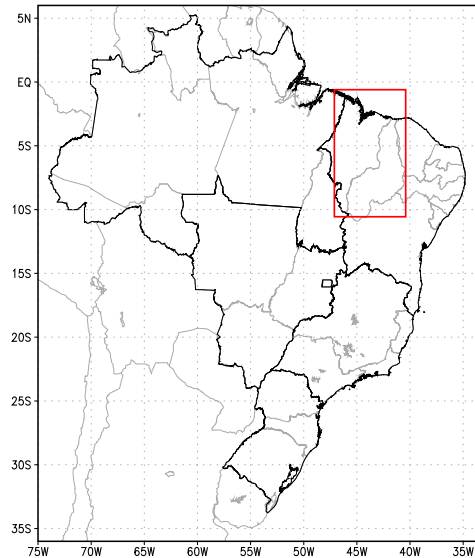


Figura 7.1 - Área de estudo: Subárea NE

Para a previsão mensal de precipitação, as redes supervisionadas MLP e a recorrente Elman foram utilizadas como modelos preditivos. Os experimentos foram efetuados usando três abordagens diferentes:

- Primeira abordagem: a NN foi treinada de forma empírica, ou seja, foi configurada durante testes preliminares, modificando os parâmetros, até atingir um resultado satisfatório;
- Segunda abordagem: a topologia da NN foi configurada automaticamente usando algoritmo de otimização (MPCA) e;
- Terceira abordagem: o algoritmo de treinamento *backpropagation* da NN foi substituído pela meta-heurística MPCA (descrito a seguir).

7.1.2 Experimentos

Para o experimento de previsão de precipitação mensal, o MPCA foi configurado para ser executado em 6 processadores, 100 iterações, com ciclo de comunicações a cada 10 iterações, 10 iterações de busca local (esquema usado para explorar a melhor solução em torno do novo local de partículas), $LI = 0,8$ e $LS = 1,2$. O critério de

parada utilizado foi o número máximo de avaliações da função objetivo (para o exemplo trabalhado: 500).

A meta-heurística MPCA é usada para gerar um conjunto de soluções candidatas, sendo que, em cada solução, a rede neural é ativada e inicia-se o processo de treinamento até que o critério de parada seja satisfeito (erro desejado ou número total de épocas). O MPCA define os parâmetros para topologia das redes supervisionadas utilizando a função objetivo definida na Equação 5.1 e o conjunto de pesos sinápticos ótimos usando a função objetivo representada na Equação 7.1.

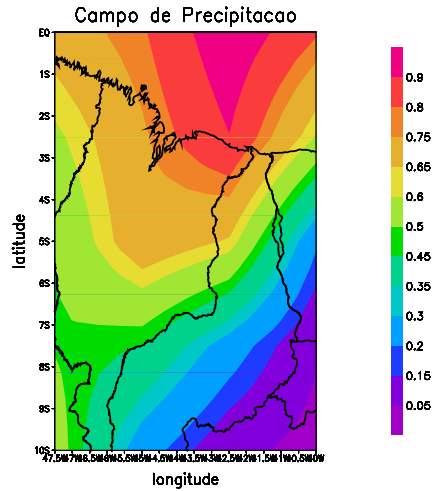
A Tabela 7.1 apresenta os parâmetros obtidos usando o algoritmo de otimização mono-objetivo MPCA das seguintes redes: NN-Empírica foi definida por experimentação empírica; a NN-MPCA, a qual teve sua topologia configurada automaticamente com o uso do MPCA; e a NN-Pesos, refere-se a rede com aprendizagem por MPCA arquitetura (pesos) obtida com o treinamento otimizado baseado em meta-heurística, ou seja, o algoritmo de aprendizagem *backpropagation* foi substituído pelo MPCA.

Tabela 7.1 - Parâmetros definidos por MPCA

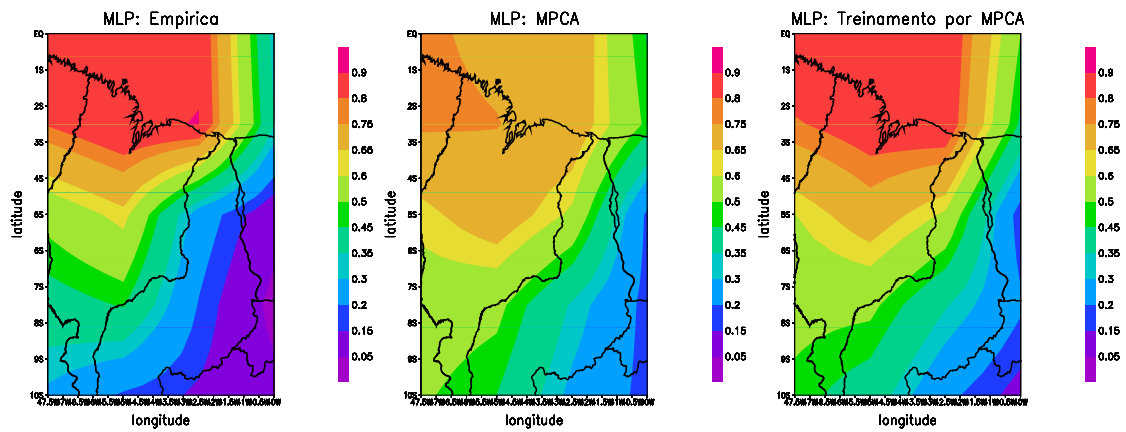
Método	n_{camadas}	$n_{\text{neurônios}}$	$n_{\text{neurônios}}$	$f_{\text{ativação}}$	η	α	EQM
NN: Empírica	1	6	-	Logística	0.4	0.6	0.000048
NN: MPCA	2	9	8	Tangente	0.5	0.2	0.000076
NN: Pesos	2	9	8	Tangente	0.5	0.2	0.000010

A Figura 7.2(a) representa o campo de observação (precipitação) extraído da base de dados de reanálise do NCEP/NCAR; a 7.2(b) mostra o resultado de previsão de precipitação, obtido com a rede MLP com a arquitetura determinada por experimentação empírica; na 7.2(c) é apresentado o resultado de previsão a partir da NN-MLP configura automaticamente pelo MPCA; a 7.2(d) é mostrado o resultado de precipitação obtido com o treinamento otimizado baseado em meta-heurística. As Figuras 7.2(e), 7.2(f) e 7.2(g) mostram os mapas de erro para a previsão das redes MLP-Empírica, MLP-MPCA e MLP-Peso, respectivamente.

A previsão de precipitação obtida com a rede recorrente Elman é mostrado nas Figuras 7.3(b), 7.3(c), 7.3(d), considerando diferentes abordagens de configuração: Elman-Empírica, a qual foi configurada empiricamente; Elman-MPCA abordagem mono-objetivo MPCA para configuração de parâmetros; e Elman-Peso, o treinamento (pesos sinápticos) foi otimizado utilizando o MPCA. As diferenças (mapas



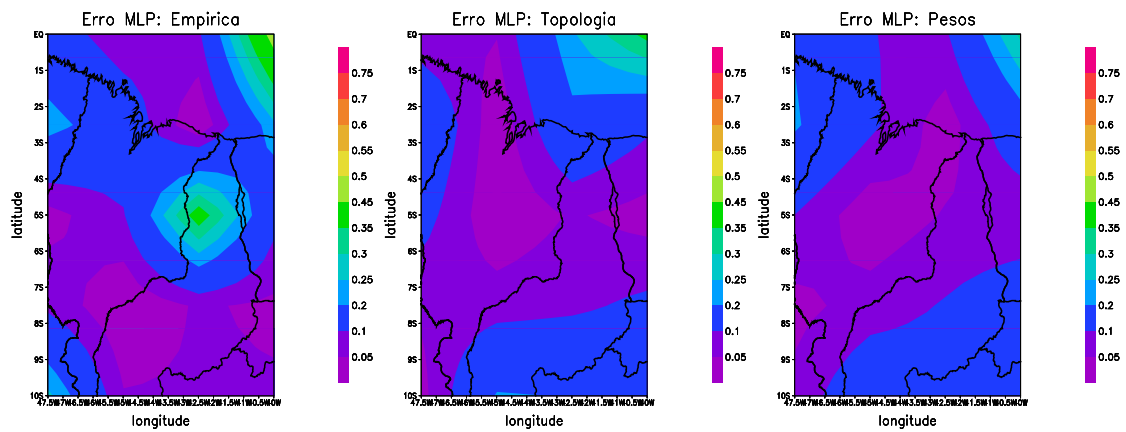
(a) Campo de precipitação de reanálise do NCEP/NCAR: Abril



(b) NN:Empírica

(c) NN:MPCA

(d) NN:Peso



(e) Erro:Empírica

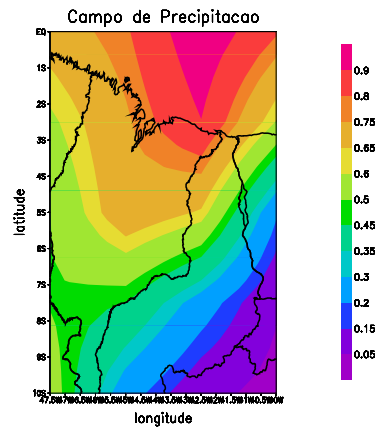
(f) Erro:MPCA

(g) Erro:Peso

Figura 7.2 - Previsão mensal de precipitação para o mês de abril – NN:MLP

de erro) entre a observação e previsão de precipitação são apresentadas nas Figuras 7.3(e), 7.3(f) e 7.3(g), para as redes Elman-Empírica, Elman-MPCA e Elman-Peso,

respectivamente.



(a) Campo de precipitação de reanálise do NCEP/NCAR: Abril

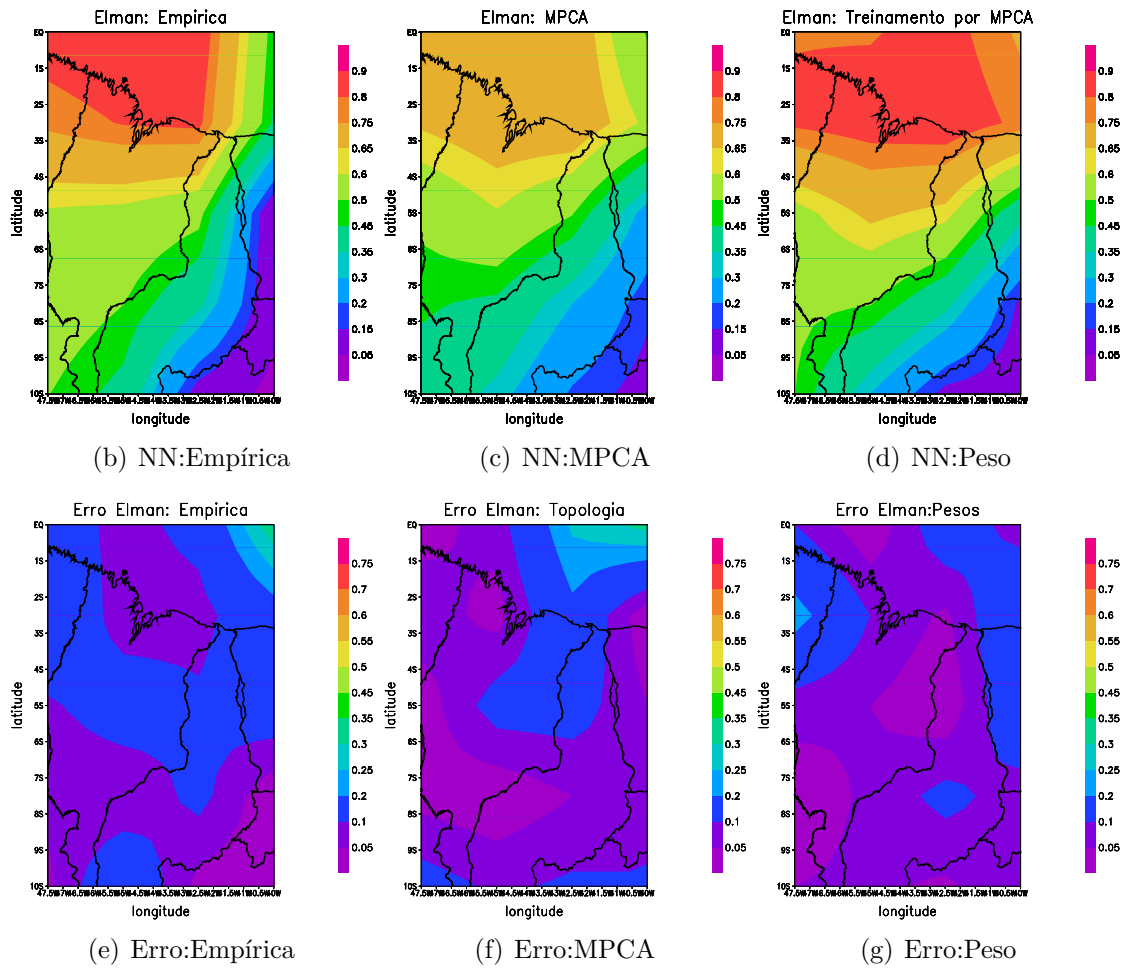


Figura 7.3 - Previsão mensal de precipitação para o mês de abril – NN:Elman

A subárea do Nordeste (que cobre parcialmente os estados Maranhão e Piauí) foi

selecionada para os testes preliminares. Para estes resultados, claramente, as redes neurais configuradas usando a meta-heurística MPCA, apresentaram os melhores resultados, em relação a NN-Empírica, a qual teve sua topologia determinada por experimentação empírica, como pode ser visto nas Figuras 7.2(e), 7.2(f), 7.3(e) e 7.3(f), para as previsões obtidas em MLP-Empírica, Elman-Empírica, MLP-MPCA e Elman-MPCA, respectivamente.

Por outro lado, o uso do MPCA aplicado como uma nova estratégia de aprendizado (treinamento), para as NNs supervisionadas MLP-Peso e Elman-Peso, apresentaram melhores resultados, comparado ao algoritmo de *backpropagation* – ver figuras 7.2(g) e 7.3(g). Considerando-se ambos os processos (autoconfiguração e treinamento), os resultados obtidos com as redes ótimas, possuem um melhor desempenho em áreas de maior concentração de população.

8 CONCLUSÕES

A presente pesquisa propôs uma metodologia que aborda o uso de meta-heurísticas na determinação de parâmetros ótimos para arquiteturas de redes neurais supervisionadas. Desta forma, o problema de definição de uma arquitetura ótima é formulado como um problema de otimização. Duas diferentes estratégias foram aplicadas à otimização: uma formulação mono-objetivo e uma formulação multiobjetivo. A otimização mono-objetivo foi solucionada pelo MPCA e o NSGA-II foi empregado para otimização multiobjetivo.

O tema sugerido neste trabalho possui considerada relevância. Redes neurais é uma área de prestígio da Inteligência Artificial e têm mostrado sua eficiência em várias áreas de aplicação, meteorologia é uma área em que redes neurais vêm ganhando cada vez mais aplicabilidade. Modelos de previsão baseados em redes neurais são de fácil uso e não requer computadores muito potentes. Desta forma, esta é uma metodologia que pode ser usada em Centros Regionais de Previsão e mesmo grandes centros operacionais de previsão de tempo e clima, como o CPTEC-INPE. Precipitação é uma variável chave para muitas aplicações da sociedade, como mencionado na [Seção 6.1](#). Assim, previsão de precipitação tem uma relevância bem estabelecida, que é um dos focos da presente tese.

As redes neurais supervisionadas MLP e Elman foram aplicadas como modelo alternativo em duas aplicações da meteorologia: uma técnica não convencional para assimilação de dados e como modelo preditivo de previsão climática de precipitação em mesoescala.

A primeira aplicação consiste em emular o filtro de Kalman para a assimilação de dados aplicado à equação da onda linear 1D. A partir dos resultados, verifica-se que as redes que tiveram seus parâmetros definidos usando as meta-heurísticas MPCA e NSGA-II, obtiveram melhor desempenho em relação a rede neural configurada empiricamente (veja a [Figura 5.2](#) e [5.3](#)). O resultado obtido com a MLP-NSGA-II apresentou menor erro durante a maior parte no domínio espacial – ver [Figura 5.4](#) em [Subsubseção 5.1.3.3](#). Uma vez treinada a rede neural o esforço computacional é menor em relação ao filtro de Kalman, o que torna a rede neural mais apta para lidar com grande volume de dados.

O segundo estudo de caso proposto neste trabalho foi o desenvolvimento de um modelo empírico de previsão climática do campo de precipitação, para as regiões Sul, Sudeste e Nordeste, a partir de dados históricos de reanálise da base de dados

do NCEP/NCAR.

Adicionalmente, foi proposto o uso de um método de redução de dados, aplicado no estudo de padrões climáticos mensais e sazonais, usando a teoria dos conjuntos aproximativos (redução de dados). A TCA foi aplicada no processo de redução de dimensionalidade de dados e permitiu a identificação de informações relevantes presentes nos dados para a previsão climática. Além disso, é uma alternativa para lidar com grandes volumes de dados, uma característica típica da meteorologia.

Os resultados obtidos para as previsões sazonais de precipitação usando a redução de dimensionalidade dos dados meteorológicos claramente apresentaram os melhores resultados, em relação as redes alimentadas com todas as variáveis disponíveis na base de dados. Em particular, na região Sul a melhora na previsão é evidente – ver Figuras 6.9(d), 6.11(d), 6.13(d) e 6.15(d) no Capítulo 6.

A arquitetura ótima determinada pelos algoritmos de otimização mostrou-se que é encontrada em um tempo menor comparado com um especialista para encontrar uma topologia aceitável. Outra vantagem é que a estratégia automática descarta a necessidade de um especialista em redes neurais tornando o emprego de redes neurais acessível a uma maior audiência.

Por fim, conclui-se que, embora a abordagem de redes neurais autoconfiguradas tenha sido aplicada em dois problemas específicos de meteorologia (assimilação de dados e previsão climática), é uma metodologia genérica e mostra-se robusta o suficiente para ser adaptada e aplicada em qualquer problema que possa ser resolvido usando redes neurais. Uma grande vantagem do uso de redes neurais é a sua implementação em hardware. Esse processamento em hardware torna o dispositivo em um neurocomputador. A implementação pode ser feita tanto em *chip* (HAMMERSTROM, 1991; RAMACHER et al., 1993), como em dispositivo reconfigurável como FPGA (*Field Programmable Gate Array*) (MUTHURAMALINGAM et al., 2008; MISRA; SAHA, 2010).

Principais contribuições do trabalho

- Configuração automática para topologia ótima de redes neurais supervisionadas usando os algoritmos de otimização MPCA e NSGA-II;
- Utilização de rede neural como uma técnica para assimilação de dados;
- Modelagem de redes neurais para previsão climática de precipitação;

- Redução da dimensão dos dados meteorológicos, uma abordagem de mineração de dados, utilizando a teoria dos conjuntos aproximativos.

Sugestões para trabalhos futuros

- Para estratégia multiobjetivo utilizar algoritmo genético epidêmico para o NSGA-II (ARAÚJO, 2014);
- Utilizar dados de saídas de modelos numéricos de previsão, como os do CPTEC-INPE;
- Desenvolver redes neurais para previsão para as regiões Norte e Centro-oeste.
- Implementar a metodologia proposta, em centros operacionais de previsão, como o CPTEC-INPE;
- Realizar a previsão climática de precipitação considerando a divisão de regiões em padrões climáticos similares, ao invés de regiões que respeitam divisões políticas;
- Utilizar outros modelos de redes neurais com aprendizagem não-supervisionada.

Publicações realizadas

- ANOCHI, J.A.; CAMPOS VELHO, H.F.; FURTADO, H.C.M.; LUZ, E.F.P. Self-configuring two types of neural networks by MPCA. *Journal of Mechanics Engineering and Automation*, v.5, p.112-120, 2015.
- ANOCHI, J.A.; CAMPOS VELHO, H.F.; Previsão climática de precipitação para a região Sul por rede neural autoconfigurada. *revista Ciência e Natura*, 2015 (aceito para publicação).
- ANOCHI, J.A.; CAMPOS VELHO, H.F.; Climate Precipitation Prediction by Neural Network. *Journal of Mathematics and System Science*, v.5, p.207-213, 2015.
- ANOCHI, J.A.; CAMPOS VELHO, H.F.; Dimensionality reduction using rough set approach for climate prediction. *Proceedings of the 12th International Conference on Applied Computing*, Dublin, Ireland, 2015.

- ANOCHI, J.A.; CAMPOS VELHO, H.F.; Previsão climática de precipitação para a região Sul por rede neural autoconfigurada. IX Workshop Brasileiro de Micrometeorologia, Santa Maria, 2015.
- ANOCHI, J.A.; CAMPOS VELHO, H.F.; FURTADO, H.C.M.; LUZ, E.F.P. Self-configuring two types neural networks by MPCA. In: Uncertainties, 2014, Rouen, France. Proceedings of the 2nd International Symposium on Uncertainty Quantification and Stochastic Modeling. Rouen, 2014. p. 429-436.
- ANOCHI, J.A.; CAMPOS VELHO, H.F. Optimization of feedforward neural network by Multiple Particle Collision Algorithm. In: 2014 IEEE Symposium on Foundations of Computational Intelligence (FOCI), 2014, Orlando. p.128-134.
- ANOCHI, J.A.; CAMPOS VELHO, H.F.; SHIGUEMORI, E.H.; LUZ, E.F.P. Data assimilation with artificial neural networks self-configuring by MPCA. In: EngOpt, 2014, Lisbon. EngOpt 2014 - International Conference on Engineering Optimization, 2014.
- ANOCHI, J.A.; CAMPOS VELHO, H.F.; SILVA, J.D.S. Neural networks in the study of climate patterns seasonal. In: CCIS, 2014, Asuncion. 3rd Conference of Computational Interdisciplinary Sciences, 2014.
- ANOCHI, J.A.; CAMPOS VELHO, H.F. Climate precipitation prediction on South of Brazil by self-configuring neural network. In: XVIII CBMET, 2014, Recife-PE. XVIII CBMET Congresso Brasileiro de Meteorologia, 2014.
- ANOCHI, J.A.; SAMBATTI, S.B.M.; LUZ, E.F.P.; CAMPOS VELHO, H.F. New learning strategy for supervised neural network: MPCA meta-heuristic approach. In: 11th Brazilian Congress (CBIC) on Computational Intelligence, 2013, Porto de Galinhas-PE. 1st BRICS Countries Congress (BRICS-CCI) and 11th Brazilian Congress (CBIC) on Computational Intelligence, 2013.
- ANOCHI, J.A.; SAMBATTI, S.B.M.; LUZ, E.F.P.; CAMPOS VELHO, H.F. Previsão Climática de Precipitação usando Rede Neural. In: XVII CBMET Congresso Brasileiro de Meteorologia, 2012, Gramado-RS. XVII CBMET Congresso Brasileiro de Meteorologia, 2012.

REFERÊNCIAS BIBLIOGRÁFICAS

ANOCHI, J. **Modelos baseados em redes neurais para o estudo de padrões climáticos sazonais a partir de dados tratados com a teoria dos conjuntos aproximativos**. 2010. 196 p. Dissertação (Mestrado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2010. 5, 9, 71, 72, 76, 79, 88, 100

ANOCHI, J.; CAMPOS VELHO, H. Optimization of feedforward neural network by multiple particle collision algorithm. In: FOUNDATIONS OF COMPUTATIONAL INTELLIGENCE, 3., 2014, Orlando, FL. **Proceedings...** Orlando: FOCI/IEEE, 2014. p. 128–134. ISBN 978-1-4799-4492-7. 38

APRIANTI, W.; MUKHLASH, I. The application of rough set and fuzzy rough set based algorithm to classify incomplete meteorological data. In: INTERNATIONAL CONFERENCE ON DATA AND SOFTWARE ENGINEERING, 10., 2014, Surabaya, Indonesia. **Proceedings...** Indonesia: ICODSE/IEEE, 2014. p. 1–6. ISBN 978-1-4799-7996-7. 45

ARAÚJO, S. A. **Calibração multiobjetivo de modelos hidrológico e de superfície atmosférico**. Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2014. 119

BABOO, S. S.; SHEREEF, I. K. An efficient weather forecasting system using artificial neural network. **International journal of environmental science and development**, v. 1, n. 4, p. 321–326, 2010. 8

BECCENERI, J. **Meta-heurística e otimização combinatória: aplicações em problemas ambientais**. São José dos Campos, 2013. 31–42 p. 21p. Mini Curso de Escola de Verão. Disponível em: <http://www.lac.inpe.br/elac13/arquivos/MiniCurso_02ELAC2012.pdf>. Acesso em: 28 jan. 2015. 27

BENARDOS, P.; VOSNIAKOS, G.-C. Optimizing feedforward artificial neural network architecture. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 20, n. 3, p. 365–382, 2007. 4, 57

BENNETT, A. **Inverse modeling of the ocean and atmosphere**. United Kingdom: Cambridge University Press, 2002. 225 p. 7, 66

BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. **ACM Computing Surveys (CSUR)**, ACM, v. 35, n. 3, p. 268–308, 2003. 32, 33

BRAGA, A.; CARVALHO, A.; LUDERMIR, T. **Redes neurais artificiais: teoria e aplicações**. 1°. ed. Rio de Janeiro, RJ: JC, 2000. 13, 14, 18, 57

CAMPOS VELHO, H.; CINTRA, R.; FURTADO, H. **Introdução à Assimilação de Dados**. março 2007. 129p. Mini Curso de Assimilação de Dados. Disponível em: <http://www.lac.inpe.br/~haroldo/Curso-DataAssimilation/Curso_Assim_Dados-INPE-2.pdf>. Acesso em: 01 jun. 2015. 64

CARVALHO, A.; RAMOS, M.; CHAVES, A. Metaheuristics for the feedforward artificial neural network (ANN) architecture optimization problem. **Neural Comput Applic**, v. 20, p. 1273–1284, 2011. 4, 57, 58, 59

CAVALCANTI, I. F.; MARENGO, J. A.; SATYAMURTY, P.; NOBRE, C. A.; TROSNIKOV, I.; BONATTI, J. P.; MANZI, A. O.; TARASOVA, T.; PEZZI, L. P.; D'ALMEIDA, C. et al. Global climatological features in a simulation using the cptec-cola agcm. **Journal of Climate**, v. 15, n. 21, p. 2965–2988, 2002. 71

CENTRO DE PREVISÃO DE TEMPO E ESTUDOS CLIMÁTICOS (CPTEC). **Início da estação chuvosa nas regiões SE/CO/S/NE do Brasil**. Cachoeira Paulista, 2015. Disponível em: <<http://energia1.cptec.inpe.br/regiaoafetadas/pt>>. Acesso em: 01 ago. 2015. 99

_____. **Supercomputação do INPE**. Cachoeira Paulista, 2015. Disponível em: <<http://supercomputacao.inpe.br/recursos2>>. Acesso em: 05 jul. 2015. 73

CINTRA, R. **Assimilação de dados com redes neurais artificiais em modelo de circulação geral da atmosfera**. Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2010. 7, 64

CINTRA, R. S.; CAMPOS VELHO, H. F. Data assimilation by artificial neural networks for an atmospheric general circulation model conventional observation. **Bulletin of the American meteorological Society**, v. 77, n. 3, p. 437–471, 2014. 7

- COELHO, C.; CARDOSO, D.; FIRPO, M. Precipitation diagnostics of an exceptionally dry event in são paulo, brazil. **Theoretical and Applied Climatology**, Springer, p. 1–16, 2015. 1
- COSTA, M.; BRAGA, A.; MENEZES, B. Improving neural networks generalization with new constructive and pruning methods. **Intelligent and Fuzzy Systems**, v. 13, p. 75–83, 2003. 4
- COSTA, M.; BRAGA, A.; MENEZES, B.; TEIXEIRA, R.; PARMA, G. Training neural networks with a multi-objective sliding mode control algorithm. **Neurocomputing**, v. 51, p. 467–476, 2003. 4
- DALEY, R. **Atmospheric data analysis**. New York: Cambridge university press, 1993. 472 p. 6
- DEB, K. **Multi-objective optimization using evolutionary algorithms**. England: John Wiley & Sons, 2001. 544 p. 30, 31
- DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. **Evolutionary Computation, IEEE Transactions on**, IEEE, v. 6, n. 2, p. 182–197, 2002. 5, 40, 41, 42, 43, 44
- DEMPSTER, A. Upper and lower probabilities induced by a multivalued mapping. **The annals of mathematical statistics**, JSTOR, p. 325–339, 1967. 45
- DORIGO, M.; BIRATTARI, M.; STUTZLE, T. Ant colony optimization. **Computational Intelligence Magazine**, IEEE, v. 1, n. 4, p. 28–39, 2006. 33
- DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: optimization by a colony of cooperating agents. **Systems, Man, and Cybernetics**, IEEE, v. 26, n. 1, p. 29–41, 1996. 26
- DORNELLES, F. **Previsão contínua de níveis fluviais com redes neurais utilizando previsão de precipitação**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, Rio Grande do Sul, 2007. 8
- DOTY, B. **Grid Analysis and Display System (GrADS)**. Virginia, 2009. Disponível em: <<http://grads.iges.org/grads/head.html>>. Acesso em: 05 jan. 2015. 75
- ECHEVARRÍA, L.; SANTIAGO, O. L.; Silva Neto, A. Aplicación de los algoritmos evolución diferencial y colisión de partículas al diagnóstico de fallos en

- sistemas industriales. **Revista Investigación Operacional**, v. 33, n. 2, p. 160–172, 2012. 37
- ELMAN, L. Finding structure in time. **Cognitive Science**, v. 14, p. 179–211, 1990. 18
- FARMER, J.; PACKARD, N.; PERELSON, A. The immune system, adaptation, and machine learning. **Physica D Nonlinear Phenomena**, Elsevier, v. 22, n. 1, p. 187–204, 1986. 33
- FAUSETT, L. V. **Fundamentals of neural networks: architectures, algorithms, and applications**. 1°. ed. Upper Saddle River, NJ, USA: Prentice-Hall Englewood Cliffs, 1994. 461 p. 13
- FUKUSHIMA, K. Cognitron: A self-organizing multilayered neural network. **Biological cybernetics**, Springer, v. 20, n. 3-4, p. 121–136, 1975. 20
- FURTADO, H. **Redes neurais para assimilação de dados em um modelo de circulação oceânica**. Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2012. 64
- FURTADO, H.; CAMPOS VELHO, H.; MACAU, E. Assimilação de dados com redes neurais artificiais em equações diferenciais. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND APPLICATIONS, 10., 2011, Águas de Lindóia, São Paulo. **Anais... Águas de Lindóia: DINCON**, 2011. p. 595–598. (DOI <http://dx.doi.org/10.5540/DINCON.2011.001.1.0152>). 7, 66, 67, 69
- GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers & operations research**, Elsevier, v. 13, n. 5, p. 533–549, 1986. 26, 32
- _____. Tabu search a tutorial. **Interfaces**, INFORMS, v. 20, n. 4, p. 74–94, 1990. 33
- GOH, C.; TAN, K. **Evolutionary multi-objective optimization in uncertain environments**. 1. ed. Berlin: Springer-Verlag Berlin Heidelberg, 2009. 271 p. ISBN 978-3-540-95976-2. 27, 29
- GOLDBERG, D. **Genetic Algorithms in Search, Optimization and Machine Learning**. 1st. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. 381 p. ISBN 0201157675. 33, 39

GORDON, N.; SALMOND, D.; SMITH, A. Novel approach to nonlinear/non-gaussian bayesian state estimation. In: IET. **IEE Proceedings F (Radar and Signal Processing)**. United Kingdom, 1993. v. 140, n. 2, p. 107–113. 7

GRIFFITH, A. **Data assimilation for numerical weather prediction using control theory**. Tese (Doutorado) — University of Reading, Inglaterra, 1997. 6

GROSSBERG, S. Birth of a learning law. **Neural Networks**, Elsevier, v. 11, n. 1, p. 1–7, 1998. 20

HAMMERSTROM, D. A highly parallel digital architecture for neural network emulation. In: Delgado-Frias, J.; Moore, W. (Ed.). **VLSI for Artificial Intelligence and Neural Networks**. New York: Springer US, 1991. cap. 5, p. 357–366. ISBN 978-1-4613-6671-3. Disponível em: <http://dx.doi.org/10.1007/978-1-4615-3752-6_35>. 118

HÄRTER, F.; CAMPOS VELHO, H. Recurrent and feedforward neural networks applied to the data assimilation in chaotic dynamics. **Revista Brasileira de Meteorologia**, v. 20, n. 3, p. 411–420, 2005. 7, 64, 67

HÄRTER, F. P.; CAMPOS VELHO, H. F. Data assimilation procedure by recurrent neural network. **Engineering Applications of Computational Fluid Mechanics**, Taylor & Francis, v. 6, n. 2, p. 224–233, 2012. 67

HAYKIN, S. **Neural networks principles and practices**. 2°. ed. Canada: Prentice Hall, 1998. 823 p. 2, 11, 13, 14, 15, 16, 17, 19, 20, 21, 22, 24, 25

HEBB, D. O. **The organization of behavior: A neuropsychological theory**. London: Psychology Press, 1949. 335 p. 11, 20

Hernández Torres, R.; LUZ, E.; CAMPOS VELHO, H. Multi-particle collision algorithm for solving an inverse radiative problem. In: CONSTANDA, C.; KIRSCH, A. (Ed.). **Integral Methods in Science and Engineering**. Springer International Publishing Switzerland, 2015. cap. 25, p. 275–284. ISBN 978-3-319-16727-5. Disponível em: <<http://www.springer.com/us/book/9783319167268>>. 38

HOLLAND, J. **Adaptation in Natural and Artificial Systems**. Cambridge, MA, USA: MIT Press, 1975. ISBN 0-262-58111-6. 39

HOPFIELD, J. Neural networks and physical systems with emergent collective computational abilities. **National academy of sciences**, National Acad Sciences, v. 79, n. 8, p. 2554–2558, 1982. 12

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS. CENTRO DE PREVISÃO DE TEMPO E ESTUDOS CLIMÁTICOS (INPE.CPTEC). **Previsão Climática**. 2015. Disponível em: <<http://clima1.cptec.inpe.br>>. Acesso em: 27 jun. 2015. 71

JAZWINSKI, A. **Stochastic processes and filtering theory**. New York: Academic Press, 1970. 376 p. 65

KALMAN, R. A new approach to linear filtering and prediction problems. **Journal of Fluids Engineering**, American Society of Mechanical Engineers, v. 82, n. 1, p. 35–45, 1960. 7, 65

KALMAN, R. E.; BUCY, R. S. New results in linear filtering and prediction theory. **Journal of Fluids Engineering**, American Society of Mechanical Engineers, v. 83, n. 1, p. 95–108, 1961. 65

KALNAY, E. **Atmospheric modeling, data assimilation, and predictability**. New York: Cambridge university press, 2003. 364 p. 6, 65

KALNAY, E.; KANAMITSU, M.; KISTLER, R.; COLLINS, W.; DEAVEN, D.; GANDIN, L.; IREDELL, M.; SAHA, S.; WHITE, G.; WOOLLEN, J. et al. The ncep/ncar 40-year reanalysis project. **Bulletin of the American meteorological Society**, v. 77, n. 3, p. 437–471, 1996. 74

KIRKPATRICK, S. Optimization by simulated annealing: Quantitative studies. **Journal of statistical physics**, Springer, v. 34, n. 5-6, p. 975–986, 1984. 26

KNUPP, D.; NETO, A.; SACCO, W. Radiative properties estimation with the particle collision algorithm based on a sensitivity analysis. **High Temperatures–High Pressures**, v. 38, p. 137–151, 2009. 34

KOHONEN, T. Self-organized formation of topologically correct feature maps. **Biological cybernetics**, Springer, v. 43, n. 1, p. 59–69, 1982. 12, 20

KOMOROWSKI, J.; ØHRN, A. Modelling prognostic power of cardiac tests using rough sets. **Artificial Intelligence in Medicine**, Elsevier, v. 15, n. 2, p. 167–191, 1999. 51, 52, 53

- KOMOROWSKI, J.; PAWLAK, Z.; POLKOWSKI, L.; SKOWRON, A. **Rough Set a tutorial**. Polond, 1998. 97 p. 47, 48, 50
- KOUSKY, V.; CHU, P. Flutuations in annual rainfall for northeast brazil. **Journal of the Meteorological Society of Japan**, v. 56, p. 457–465, 1978. 79
- KOVACS, Z. **Redes neurais artificiais: fundamentos e aplicações**. 2°. ed. São Paulo: Editora Acadêmica, 2002. 14
- LECUN, Y. Une procedure d'apprentissage pour reseau a seuil assymetrique. **Cognitiva**, v. 85, p. 599–604, 1985. 12
- LEVENBERG, K. A method for the solution of certain non-linear problems in least squares. **Quarterly of Applied Mathematics**, v. 2, n. 1, p. 164–168, 1944. 26
- LUZ, E. **Meta-heurísticas paralelas na solução de problemas inversos**. Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2012. Disponível em: <<http://urlib.net/sid.inpe.br/mtc-m19/2012/02.22.17.13>>. 35, 36, 37, 38
- LUZ, E.; BECCENERI, J.; CAMPOS VELHO, H. A new multi-particle collision algorithm for otimization in a high-performance environment. **Journal of Computacional Interdisciplinary Sciences**, v. 1, p. 3–10, 2008. 5, 37
- MARQUARDT, D. An algorithm for least-squares estimation of nonlinear parameters. **Journal of the Society for Industrial & Applied Mathematics**, SIAM, v. 11, n. 2, p. 431–441, 1963. 26, 111
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943. 11, 14, 15
- METROPOLIS, N.; RESENBLUTH, A.; ROSENBLUTH, M.; TELLER, A.; TELLER, E. Equation of state calculations by fast computing machines. **The Journal of Chemical Physics**, v. 6, n. 21, p. 1087–1092, 1953. 34, 37
- MINSKY, M.; PAPERT, S. **Perceptron: an introduction to computational geometry**. United States of America: The MIT Press, Cambridge, expanded edition, 1969. 308 p. 11, 12
- MISRA, J.; SAHA, I. Artificial neural networks in hardware: A survey of two decades of progress. **Neurocomputing**, Elsevier, v. 74, n. 1, p. 239–255, 2010. 118

MUTHURAMALINGAM, A.; HIMAVATHI, S.; SRINIVASAN, E. Neural network implementation using fpga. **International journal of information technology**, v. 4, n. 2, p. 86–92, 2008. 118

NACIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION (NOAA). **Reanalysis Datasets at PSD**. 2012. Reanalysis Datasets. Disponível em: <<http://www.esrl.noaa.gov/psd/>>. Acesso em: 17 jan. 2012. 73

NOWOSAD, A.; CAMPOS VELHO, H.; NETO, A. R. Neural network as a new approach for data assimilation. In: CONGRESSO BRASILEIRO DE METEOROLOGIA, 11., 2000, Rio de Janeiro. **Proceedings...** Rio de Janeiro: CBMET, 2000. p. 3078–3086. 7, 64

ØHRN, A. **Discernibility and rough sets in medicine: tools and applications**. Tese (Doutorado) — Norwegian University of Science and Technology (NTNU), Norway, 1999. 53, 54, 55

ØHRN, A. **ROSETTA technical reference manual**. Norway, Novembro 1999. 1–66 p. Disponível em: <<http://www.lcb.uu.se>>. 46, 53, 54, 55, 56

OLIVEIRA, A. S. **Interações entre sistemas frontais na América do Sul e convecção na Amazônia**. 139p. (INPE-4008-TDL/239). Dissertação (Mestrado em Meteorologia) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 1986. 87

PAL, S.; SKOWRON, A. (Ed.). **Rough-Fuzzy Hybridization A New Trend in Decision Making**. 1. ed. Secaucus, USA: Springer-Verlag, 1999. 454 p. 47

PARKER, D. **Learning Logic: Casting the Cortex of the Human Brain in Silicon**. Massachusetts Institute of Technology, Center for Computational Research in Economics and Management Science, 1985. 47 p. Disponível em: <<http://books.google.com.br/books?id=2kS9GwAACAAJ>>. Acesso em: 03 maio 2015. 12

PAWLAK, Z. Rough sets. **International Journal of Computer & Information Sciences**, Springer, v. 11, n. 5, p. 341–356, 1982. 45, 46, 48, 51

PESSOA, A.; STEPHANY, S. An innovative approach for attribute reduction in rough set theory. **Intelligent Information Management**, Scientific Research Publishing, v. 6, n. 5, p. 223–239, 2014. 45

PETERS, J. F.; SURAJ, Z.; SHAN, S.; RAMANNA, S.; PEDRYCZ, W.; PIZZI, N. Classification of meteorological volumetric radar data using rough set methods. **Pattern Recognition Letters**, Elsevier, v. 24, n. 6, p. 911–920, 2003. 45

PITTS, W.; MCCULLOCH, W. S. How we know universals the perception of auditory and visual forms. **The Bulletin of mathematical biophysics**, Springer, v. 9, n. 3, p. 127–147, 1947. 11

PRECHELT, L. et al. Proben1: A set of neural network benchmark problems and benchmarking rules. Citeseer, 1994. 2

QUADRO, M.; MACHADO, L.; CALBETE, S.; BATISTA, N.; OLIVEIRA, G. **Climanálise: Boletim de Monitoramento e Análise Climática**. São José dos Campos: INPE, v. 9, n. 1, jan. 1996. Disponível em: <<http://climanalise.cptec.inpe.br/~rcliman1/boletim/cliesp10a/chuesp.html>>. Acesso em: 3 mai. 2015. 99

RAMACHER, U.; RAAB, W.; ANLAUF, J.; HACHMANN, U.; BEICHTER, J.; BRÜLS, N.; WESSELING, M.; SICHENEDER, E.; MÄNNER, R.; GLÄSS, J. Multiprocessor and memory architecture of the neurocomputer synapse-1. **International journal of neural systems**, World Scientific, v. 4, n. 4, p. 333–336, 1993. 118

REED, R. Pruning algorithms-a survey. **Transactions Neural Networks**, IEEE Press, Piscataway, NJ, USA, v. 4, n. 5, p. 740–747, 1993. ISSN 1045-9227. Disponível em: <<http://dx.doi.org/10.1109/72.248452>>. 3

REEVES, C. **Modern heuristic techniques for combinatorial problems**. New York, USA: John Wiley & Sons, Inc., 1993. 320 p. 32

ROCHA, H. P. **Abordagens multi-objetivo para o treinamento de redes neurais e seleção de características**. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte - Minas Gerais, Março 2012. Disponível em: <<http://hdl.handle.net/1843/BUOS-8T2HBR>>. 3

ROPELEWSKI, C.; HALPERT, M. Precipitation patterns associated with the high index phase of the southern oscillation. **Journal of climate**, v. 2, n. 3, p. 268–284, 1989. 72

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958. 11, 21

- RUIVO, H.; CAMPOS VELHO, H.; SAMPAIO, G.; RAMOS, F. Analysis of extreme precipitation events using a novel data mining approach. **American Journal of Environmental Engineering**, v. 5, n. 1A, p. 96–105, 2015. 8, 71
- RUMELHART, D. E.; MCCLELLAND, J. L. **Parallel distributed processing: Explorations in the microstructures of cognition**. 1°. ed. Cambridge, MA, USA: MIT Press, 1986. 576 p. 12
- RUMMELHART, D.; HINTON, G.; WILLIAMS, R. Learning representations by back-propagation errors. **Nature**, v. 323, p. 533–536, 1986. 12
- SACCO, W.; LAPA, C.; PEREIRA, C.; FILHO, H. Two stochastic optimization algorithms applied to nuclear reactor core design. **Progress in Nuclear Energy**, v. 48, p. 525–539, 2006. 34
- _____. A metropolis algorithm applied to a nuclear power plant auxiliary feedwater system surveillance tests policy optimization. **Progress in Nuclear Energy**, v. 50, p. 15–21, 2008. 34
- SACCO, W.; OLIVEIRA, C. A new stochastic optimization algorithm based on a particle collision metaheuristic. In: WORLD CONGRESS OF STRUCTURAL AND MULTIDISCIPLINARY OPTIMIZATION, 6., 2005, Rio de Janeiro, Brazil. **Proceedings...** Rio de Janeiro: WCSMO, 2005. p. 1–8. 34, 37
- SAMBATTI, S.; ANOCHI, J.; LUZ, E.; A.R.CARVALHO; SHIGUEMORI, E.; CAMPOS VELHO, H. Automatic configuration for neural network applied to atmospheric temperature profile identification. In: INTERNATIONAL CONFERENCE ON INTERNATIONAL CONFERENCE ON ENGINEERING OPTIMIZATION, 3., 2012, Rio de Janeiro, Brazil. **Proceedings...** Rio de Janeiro: EngOpt, 2014. p. 1–6. 37
- SAMPAIO, G.; SILVA DIAS, P. Evolução dos modelos climáticos e de previsão de tempo e clima. **Revista USP**, n. 103, p. 41–54, 2014. 1, 72
- SESHADRI, A. **Multi-objective optimization using evolutionary algorithms (MOEA)**: Nsga-ii: A multi-objective optimization algorithm. 2006. Disponível em: <<http://www.mathworks.com/matlabcentral/fileexchange/10429>>. Acesso em: 11 março 2015. 62
- SINGIRESU, R. **Engineering optimization theory and practice**. 4°. ed. Canada: John Wiley, 2009. 803 p. 27

SOUSA, W.; SOUSA, F. Rede neural artificial aplicada à previsão de vazão da bacia hidrográfica do rio piacó. **Revista Brasileira de Engenharia Agrícola e Ambiental**, SciELO Brasil, v. 14, n. 2, p. 173–180, 2010. 8, 71

SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. **Evolutionary computation**, MIT Press, v. 2, n. 3, p. 221–248, 1994. 40

STONE, M. Cross-validation: A review. **Mathematische Operationforsch Statischen**, v. 9, p. 127–139, 1978. 3

TALAGRAND, O.; COURTIER, P. Variational assimilation of meteorological observations with the adjoint vorticity equation. i: Theory. **Quarterly Journal of the Royal Meteorological Society**, Wiley Online Library, v. 113, n. 478, p. 1311–1328, 1987. 7

TEIXEIRA, R.; BRAGA, A.; TAKAHASHI, R.; SALDANHA, R. Improving generalization of mlp with multi-objective optimization. **Neuro**, v. 35, p. 189–194, 2000. 3

VIANELLO, R. L.; ALVES, A. **Meteorologia básica e Aplicações**. 4^o. ed. Viçosa: UFV, 2006. 449 p. 72

WEIGEND, A.; HUBERMAN, B. Predicting the future: a connectionist approach. **International Journal of Neural Systems**, v. 1, n. 3, p. 193–209, 1990. 3

WERBOS, P. **Beyond regression: New tools for prediction and analysis in the behavioral sciences**. Tese (Doutorado) — Harvard University, Cambridge MA, August 1974. 12

WIDROW, B. Generalization and information storage in network of adaline neurons. **Self-organizing systems**, Spartan Books, v. 1, p. 435–462, 1962. 11, 20

WILLIAMS, D. R. G. H. R.; HINTON, G. Learning representations by back-propagating errors. **Nature**, London, p. 323–533, 1986. 26, 111

YANG, X.; KOZIEL, S. **Computational optimization and applications in engineering and industry**. Berlin: Springer Berlin Heidelberg, 2011. 271 p. 33

YANG, X.-S. A new metaheuristic bat-inspired algorithm. In: **Nature inspired cooperative strategies for optimization (NICSO 2010)**. London: Elsevier, 2010. v. 284, p. 65–74. 33

YAO, X. Evolving artificial neural networks. **Proceedings of the IEEE**, IEEE, v. 87, n. 9, p. 1423–1447, 1999. 26

ZADEH, L. Fuzzy sets. **Information and Control**, Elsevier, v. 8, p. 338–353, 1965. 45

_____. Fuzzy sets as a basis for a theory of possibility. **Fuzzy sets and systems**, Elsevier, v. 1, p. 3–28, 1978. 45

APÊNDICE A - SISTEMA ROSETTA

A ferramenta Rosetta trabalha sob plataformas de 32 bits e oferece um ambiente no qual o usuário de um modo simples pode ver e acompanhar os itens de dados individuais em um projeto de análise. Alguns recursos computacionais oferecidos pelo sistema Rosetta incluem:

Importação e exportação

- Integração parcial com SGBDs pertinentes à interface ODBC (*Open Database Connectivity*), o que permite a análise de dados em diferentes formatos.
- Exportação de regras, redutos, tabelas, gráficos e outros objetos para vários formatos, incluindo XML, C++, arquivo de texto ASCII, Matlab e Prolog, proporcionando assim a integração com outros tipos de ferramentas para análise de dados.

Pré-processamento

- Compreende na limpeza dos dados, ou seja, a remoção de registros de dados incompletos ou ausentes presentes na base de dados.
- Discretização de atributos numéricos.

Computação

- Apoio à aprendizagem supervisionada e não supervisionada.
- Suporte para conceitos definidos pelo usuário de discernibilidade.
- Computação eficiente dos redutos exatos ou aproximados.
- Geração de regras (se-então) ou padrões descritivos via redutos.
- Execução de arquivos de *scripts*.
- Suporte para validação cruzada.

Pós-Processamento

- Filtragem avançada de conjuntos de redutos e regras baseada no suporte, tamanho e qualidade. A qualidade pode ser avaliada segundo algumas medidas nomeadas no Rosetta,

Validação e análise

- Aplicação das regras sintetizadas para exemplos invisíveis.
- Geração de matrizes de confusão, curvas ROC e curvas de calibração.
- Avaliação das regras individuais de acordo com as medidas de qualidade.
- Utilitários para testes de hipóteses.

Outros

- Agrupamento via relações de tolerância.
- Computação de divisórias e de precisão da variável de aproximação.
- Suporte para amostragem aleatória de observações.
- Código-fonte aberto.
- Documentação.

Suporte a árvores

Cada item em um projeto de análise de dados é representado por seu próprio ícone específico para seu tipo e cada projeto organiza estes ícones em uma árvore. A topologia da árvore exhibe como os itens de dados, relacionam-se entre si, de um modo intuitivo e imediato. Um exemplo de uma árvore de projeto é mostrado na [Figura A.1](#).

Visualização de dados

Podem ser visualizados todos os itens de dados das árvores de projeto em janelas individuais, tipicamente em visões de tabelas. Um exemplo de uma área de trabalho é mostrado em [Figura A.2](#).

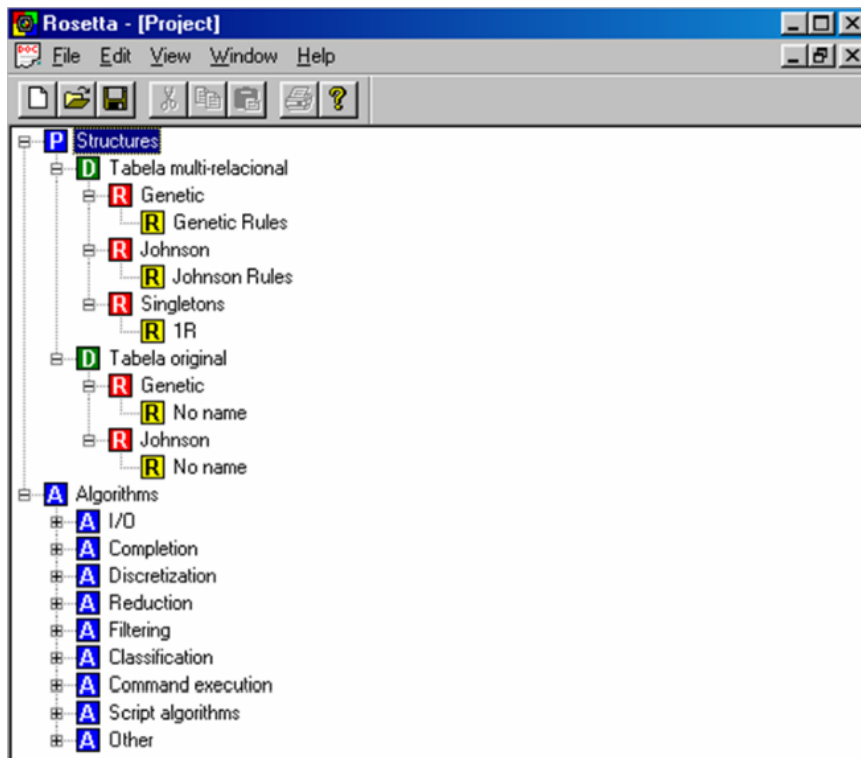


Figura A.1 - Exemplo de um projeto no Rosetta

Genetic Rules

	Rule	LHS Suppor	RHS Suppor	RHS Accura	LHS Cov
1	ZIP(127) AND 3	3	3	1.0	0.000515
2	ZIP(60) AND 3	3	3	1.0	0.000515
3	PLEVEN(4) A 3	3	3	1.0	0.000515
4	ZIP(127) AND 3	3	3	1.0	0.000515
5	ZIP(60) AND 3	3	3	1.0	0.000515
6	ZIP(127) AND 3	3	3	1.0	0.000515
7	ZIP(60) AND 3	3	3	1.0	0.000515
8	ZIP(127) AND 3	3	3	1.0	0.000515
9	ZIP(60) AND 3	3	3	1.0	0.000515
10	PPLEZIER(1) A 3	3	3	1.0	0.000515
11	ZIP(60) AND 3	3	3	1.0	0.000515

1R

	Rule	LHS Suppor	RHS Suppor	RHS Accura	LHS Cov
1	PWVERKT(0) = 5801	5453, 348	5453, 348	0.94001, 0.05	0.996393
2	AVRAAUT(0) 5813	5465, 348	5465, 348	0.940134, 0.0	0.998454
3	PVRAAUT(0) 5813	5465, 348	5465, 348	0.940134, 0.0	0.998454
4	AWVERKT(0) = 5801	5453, 348	5453, 348	0.94001, 0.05	0.996393
5	PZEILPL(0) = 5819	5472, 347	5472, 347	0.940368, 0.0	0.999485
6	AZEILPL(0) = 5819	5472, 347	5472, 347	0.940368, 0.0	0.999485
7	APERSONG(0) 5791	5444, 347	5444, 347	0.940079, 0.0	0.994675
8	PPERSONG(0) 5791	5444, 347	5444, 347	0.940079, 0.0	0.994675
9	PWAOREG(0) 5799	5455, 344	5455, 344	0.940679, 0.0	0.996049
10	AWAOREG(0) 5799	5455, 344	5455, 344	0.940679, 0.0	0.996049
11	PPERSONG(0) 5789	5454, 335	5454, 335	0.942132, 0.0	0.994332

Johnson Rules

	Rule
1	ZIP(127) AND PPERSONG(6) AND PBRAND(4) => CARAVAN(1)
2	ZIP(749) AND PPERSONG(6) => CARAVAN(1)
3	ZIP(362) AND PPERSONG(6) => CARAVAN(0) OR CARAVAN(1)
4	ZIP(394) AND PPERSONG(6) => CARAVAN(1)
5	ZIP(81) AND PWAPART(0) AND PPERSONG(6) => CARAVAN(1) O
6	ZIP(13) AND PWAPART(2) AND PPERSONG(6) AND PBRAND(4) AN
7	ZIP(517) AND PPERSONG(6) => CARAVAN(1)
8	ZIP(60) AND PWAPART(2) AND PPERSONG(6) AND PBROM(0) =>
9	ZIP(101) AND PWAPART(2) => CARAVAN(1)
10	ZIP(404) AND PPERSONG(6) => CARAVAN(1)
11	ZIP(742) AND PWAPART(0) => CARAVAN(1)

Figura A.2 - Exemplo de uma área de trabalho do sistema Rosetta