



UNIVERSIDADE
ESTADUAL DE LONDRINA

FELIPPE ALEX SCHEIDT

**MODELAGEM CHUVA-VAZÃO UTILIZANDO REDES
NEURAS ARTIFICIAIS**

Londrina
2010

FELIPPE ALEX SCHEIDT

**MODELAGEM CHUVA-VAZÃO UTILIZANDO REDES
NEURAS ARTIFICIAIS**

Dissertação apresentada ao Programa de Mestrado em Ciências da Computação da Universidade Estadual de Londrina, como requisito parcial à obtenção do título de Mestre.

Orientadora: Profa. Dra. Maria Angélica de Oliveira Camargo Brunetto.

Co-orientador: Prof. Dr. Glaucio Roloff.

Londrina
2010

Dados Internacionais de Catalogação-na-Publicação (CIP)

S318m Scheidt, Felipe Alex.

Modelagem chuva-vazão utilizando redes neurais artificiais / Felipe Alex Scheidt. – Londrina, 2010.
115 f. : il.

Orientador: Maria Angélica de Oliveira Camargo Brunetto..

Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Centro de Ciências Exatas e da Terra, Programa de Pós-Graduação em Ciência da Computação, 2010.

Inclui bibliografia.

1. Redes neurais artificiais – Teses. 2. Algoritmos genéticos – Teses. 3. Bacias hidrográficas – Teses. 4. Modelos ARMA – Teses. 5. Modelos híbridos – Teses. 6. Modelagem chuva-vazão – Teses. I. Brunetto, Maria Angélica de Oliveira Camargo. II. Universidade Estadual de Londrina. Centro de Ciências Exatas e da Terra. Programa de Pós-Graduação em Ciências da Computação. III. Título.

CDD: 004.032.26

FELIPPE ALEX SCHEIDT

**MODELAGEM CHUVA-VAZÃO UTILIZANDO REDES
NEURAS ARTIFICIAIS**

Dissertação apresentada ao Programa de Mestrado em Ciências da Computação da Universidade Estadual de Londrina, como requisito parcial à obtenção do título de Mestre.

COMISSÃO EXAMINADORA

Profa. Dra. Maria Angélica de Oliveira
Camargo Brunetto
Universidade Estadual de Londrina

Prof. Dr. André Carlos Ponce de Leon
Ferreira de Carvalho
ICMC-USP/São Carlos

Prof. Dr. Pedro Paulo da Silva Ayrosa
Universidade Estadual de Londrina

Prof. Dr. Robinson Samuel Vieira Hoto
Universidade Estadual de Londrina

Londrina, 27 de agosto de 2010.

*Dedico aos meus pais ...
Tamara Tscherdantzen Scheidt e
Paulo Scheidt.*

AGRADECIMENTOS

Agradeço o apoio e incentivo da professora Maria Angélica, pela orientação desde o início, mesmo nos momentos mais difíceis, com sua destreza nas observações e direcionamentos pontuais no desenvolvimento deste trabalho. Também agradeço a Fundação Parque Tecnológico Itaipu, pelo auxílio financeiro concedido por bolsa de pesquisa, de vital importância para a conclusão deste trabalho. Ao incentivo do professor Glaucio Roloff, no apoio ao desenvolvimento da temática do trabalho e no investimento de um projeto de pesquisa na área hidrológica. A minha parceira Laiza Pâmela, pela compreensão e ajuda constante durante todo o período de desenvolvimento do mestrado. E ao corpo docente do curso de mestrado de Ciências da computação da UEL pelo aprendizado e conhecimento transmitido e pela oportunidade única de participação neste curso.

*Há um livro em cada um de nós
(Clarice Lispector)*

SCHEIDT, Felipe Alex. **Modelagem Chuva-Vazão utilizando redes neurais artificiais**. 2010. 115 fl. Dissertação de Mestrado (Ciências da Computação) – Universidade Estadual de Londrina, Londrina, 2010.

RESUMO

Este trabalho apresenta uma metodologia para modelagem da relação da transformação chuva-vazão de uma bacia hidrográfica utilizando redes neurais artificiais (RNA) acoplada a um algoritmo genético (AG) num intervalo de discretização diário e mensal. Esta modelagem foi aplicada na bacia hidrográfica do rio Piquiri, localizada no estado do Paraná, Brasil. Os resultados dessa modelagem foram comparados a um modelo autorregressivo de média móvel e demonstraram que as RNAs são mais adequadas pra a modelagem da relação chuva-vazão do que os modelos autorregressivos. Além disso comparou-se o desempenho e características das redes neurais artificiais com um modelo híbrido utilizando RNA e AG, onde foi observado que o modelo híbrido obteve melhor representação do processo de transformação chuva-vazão.

Palavras-chave: Redes neurais artificiais. Algoritmos genéticos. Modelos ARMA. Bacias hidrográficas. Modelagem chuva-vazão. Modelo híbrido.

SCHEIDT, Felipe Alex. **Rainfall-runoff modeling with artificial neural networks.** 2010. 115 fl. Master thesis Research Project (Computer Science) – State University of Londrina, Londrina. 2010.

ABSTRACT

In this work it is proposed a methodology for modeling the rainfall-runoff relationship of a specific watershed, through artificial neural networks (ANN) and genetic algorithm. This model was developed on events based on daily and monthly observations. The study case is a watershed of the state of Paraná, Brasil, named Piquiri river Basin. The results are compared with a autorregressive and moving average model, and showed that neural networks has superior capacity to represent the rainfall-runoff relationship. Besides, the ANN methodology was compared with a hybrid model, which coupling ANN with genetic algorithm, and the results showed that the hybrid model fitted better than ANN, when representing the rainfall-runoff transformation.

Key words: Artificial neural networks. Genetic Algorithm. Autorregressive models. Rainfall-runoff modeling. Hybrid model.

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Ciclo hidrológico	28
Figura 2.2 – Bacia hidrográfica.....	30
Figura 2.3 – Hidrograma com médias diárias, do rio Baron Fork, AR, EUA	32
Figura 2.4 – Uma série temporal e seu respectivo correlograma	42
Figura 2.5 – Visão esquemática de um neurônio biológico	47
Figura 2.6 – Modelo de um neurônio artificial.....	49
Figura 2.7 – Função sigmóide	50
Figura 2.8 – Uma rede MLP com 1 camada oculta	57
Figura 2.9 – Aprendizagem supervisionada	59
Figura 2.10 – Exemplo de cromossomo com codificação binária.....	68
Figura 2.11 – Exemplo de cromossomo com codificação por valor.....	68
Figura 2.12 – Exemplo de cromossomo com codificação por permutação	68
Figura 2.13 – Recombinação entre dois cromossomos.....	69
Figura 2.14 – Recombinação usando dois pontos de cortes.....	70
Figura 2.15 – Mutação.....	71
Figura 2.16 – Rede neural, acoplada a um AG	73
Figura 3.1 – Bacia do rio Piquiri.....	76
Figura 3.2 – Bacia do rio Piquiri – Afluentes.....	77
Figura 3.3 – Hidrograma do rio Piquiri – medições diárias – Período: 2000.....	78
Figura 3.4 – Hidrograma do rio Piquiri – medições diárias – Período: 2001.....	78
Figura 3.5 – Hidrograma do rio Piquiri – medições diárias – Período: 2002.....	78
Figura 3.6 – Hidrograma do rio Piquiri – medições diárias – Período: 2003.....	79
Figura 3.7 – Hidrograma do rio Piquiri – medições diárias – Período: 2004.....	79
Figura 3.8 – Hidrograma do rio Piquiri – medições diárias – Período: 2005.....	79
Figura 3.9 – Hidrograma do rio Piquiri – medições diárias – Período: 2006.....	80
Figura 3.10 – Hidrograma do rio Piquiri – medições diárias – Período: 2007	80
Figura 3.11 – Hidrograma do rio Piquiri – medições diárias – Período: 2008.....	80
Figura 3.12 – Método da parada antecipada	85
Figura 3.13 – Seleção de um modelo de RNA	86
Figura 3.14 – Número de nodos ocultos e os respectivos erros.....	87
Figura 3.15 – Recombinação com dois pontos de corte.....	89

Figura 3.16 – Visão geral da arquitetura do sistema	94
Figura 4.1 – Comparação entre vazão observada e calculada (AG-RNA)	102
Figura 4.2 – Coeficiente R2 – AG-RNA	103
Figura 4.3 – Comparação entre vazão observada e calculada (RNA).....	103
Figura 4.4 – Coeficiente R2 - RNA	104
Figura 4.5 – Comparação entre vazão observada e calculada (AG-RNA)	104
Figura 4.6 – Coeficiente R2 – AG-RNA	105
Figura 4.7 – Comparação entre vazão observada e calculada usando RNA	105
Figura 4.8 – Coeficiente R2 - RNA	107
Figura 4.9 – Vazão prevista pelo modelo ARMA(10,5).....	107
Figura 4.10 – Coeficiente R2 – modelo ARMA(10,5).....	107

LISTA DE QUADROS

Quadro 1.1 – Estudos apresentados e respectivas configurações utilizadas	26
Quadro 2.1 – Exemplos de funções de transferência	50
Quadro 3.1 – Índices de correlações da estação 64767000	81
Quadro 3.2 – Séries temporais utilizadas	82
Quadro 3.3 – Modelos de entrada de dados da RNA.....	92
Quadro 4.1 – Parâmetros utilizados nas simulações AG-RNA-Diário	97
Quadro 4.2 – Desempenho dos modelos usando AG-RNA	98
Quadro 4.3 – Desempenho dos modelos usando RNA.....	99
Quadro 4.4 – Parâmetros utilizados na simulação AG-RNA-Mensal.....	100
Quadro 4.5 – Desempenho dos modelos usando AG-RNA	100
Quadro 4.6 – Desempenho dos modelos usando RNA.....	101
Quadro 4.7 – Comparação entre RNA e AG-RNA.....	101
Quadro 4.8 – Resultados obtidos	106
Quadro 4.9 – Comparação entre RNA e AG-RNA.....	108

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmos Genéticos.
AG-RNA	Algoritmos genéticos acoplados a redes neurais artificiais.
AIC	<i>Akaike information Criterion.</i>
ANA	Agência Nacional de Águas.
ANEEL	Agência Nacional de Energia Elétrica.
API	<i>Application Programming Interface.</i>
AR	<i>Autorregressive</i>
ARMA	<i>Autorregressive and moving average</i>
BP	<i>Backpropagation</i>
IPHMEN	Modelo do Instituto de Pesquisas Hidrológicas com intervalo de tempo mensal.
LMS	<i>Least mean square.</i>
MA	<i>Moving average.</i>
MLP	<i>Multi-layer perception.</i>
MSE	<i>Mean square error.</i>
NS	<i>Nash Sutcliffe.</i>
RMSE	<i>Root mean square error.</i>
RNA	Redes neurais artificiais.
SMA	<i>Simple moving average.</i>
SVM	<i>Support vector machine.</i>
SUDERSHA	Superintendência de Desenvolvimento de Recursos Hídricos e Saneamento Ambiental.
VCM	Validação cruzada múltipla

LISTA DE SÍMBOLOS

$W = (w_{ij})$ – matriz de pesos.

y_j – saída ou ativação do neurônio j .

y_j^d – saída desejada do neurônio j .

α – taxa de aprendizado.

Δw – percentual de ajuste na fórmula de atualização dos pesos sinápticos

u_j – soma ponderada de todas as entradas do neurônio j .

u_k – soma ponderada de todas as entradas do neurônio k .

$x(n)$ – vetor sinal ou vetor entrada k .

$\varphi(\cdot)$ – função de ativação.

X_j – vetor de entrada

e – erro gerado pela diferença entre a saída desejada e a saída calculada

x_0 – viés ou bias.

x_t – observação de uma série temporal num instante t .

r – coeficiente de correlação

R^2 – coeficiente de Nash-stucliffe.

X_{norm} – vetor entrada normalizado entre [0.1:0.9]

P_t – observação de precipitação num instante t .

P_m – precipitação média.

Q_t – observação de vazão num instante t

\hat{Q}_i – vazão calculada

\overline{Q}_i – vazão média

T_i – componente tendência

C_t – componente cíclica

S_t – componente de sazonalidade

R_t – componente de aleatoriedade..

Z_{pm} – série histórica média dos índice pluviométricos

Z_q – série histórica de vazão.

Z_t – série histórica de vazão e chuva desazonalizada.

SUMÁRIO

1 INTRODUÇÃO	16
1.1 OBJETIVOS	17
1.2 OBJETIVOS ESPECÍFICOS	18
1.3 JUSTIFICATIVA	19
1.4 ESTRUTURA DO TRABALHO	20
1.4.1 Fundamentação Teórica.....	20
1.4.2 Metodologia.....	20
1.4.3 Resultados e Discussão	21
1.5 REVISÃO BIBLIOGRÁFICA	21
1.6 SOBRE ESTE TRABALHO.....	25
2 FUNDAMENTOS TEÓRICOS	27
2.1 MODELOS HIDROLÓGICOS	27
2.1.1 Ciclo hidrológico	27
2.1.2 Bacia hidrográfica.....	29
2.1.3 Precipitação.....	30
2.1.4 Evaporação	31
2.1.5 Vazão.....	31
2.2 ANÁLISE DE SÉRIES TEMPORAIS.....	33
2.2.1 Introdução	33
2.2.2 Métodos estatísticos.....	37
2.2.3 Previsão... ..	37
2.2.4 Média Móvel (MA)	37
2.2.5 Filtros.....	38
2.2.6 Sazonalidade.....	38
2.2.7 Autocorrelação	39
2.2.8 Correlograma	41
2.2.9 Modelos probabilísticos	42
2.2.10 Modelos autorregressivos	43
2.2.11 ARMA – Modelo autorregressivo e média móvel	45

2.3 REDES NEURAIS	46
2.3.1 Neurônio Biológico	46
2.3.2 Neurônio Artificial	48
2.3.3 Estrutura e aplicações	51
2.3.4 Arquitetura	52
2.3.5 Aprendizagem	53
2.3.6 Perceptron	54
2.3.7 Multi-layer perceptron	55
2.3.8 Algoritmo Backpropagation	61
2.4 ALGORITMOS GENÉTICOS	66
2.4.1 Cromossomo	67
2.4.1 Seleção.....	69
2.4.1 Recombinação	69
2.4.1 Mutação.....	70
2.4.1 Algoritmo.....	71
2.5 MODELOS HÍBRIDOS	72
3 METODOLOGIA	75
3.1 ESTUDO DE CASO	75
3.1.1 Seleção dos dados para processamento	80
3.1.2 Pré-processamento	82
3.1.3 Divisão dos conjuntos.....	82
3.1.4 Método da parada antecipada.....	83
3.1.5 Definição do número de neurônios da camada oculta	85
3.1.6 Definição dos pesos iniciais usando AG.....	87
3.1.7 Modelagem das RNAs.....	90
3.1.8 Modelo ARMA	92
3.1.9 Ferramentas de desenvolvimento	93
4 RESULTADOS E DISCUSSÃO	95
4.1 ANÁLISE DOS RESULTADOS	96
4.1.1 Análise preliminar	96
4.1.2 Modelos AG-RNA em escala diária	96
4.1.3 Modelos RNA em escala diária	98

4.1.4 Modelos AG-RNA em escala mensal	99
4.1.5 Modelos RNA em escala mensal	101
4.1.6 Comparação das técnicas – Fase de validação	101
4.1.7 Fase de testes	102
5 CONCLUSÕES E RECOMENDAÇÕES	110
REFERÊNCIAS.....	112

1 INTRODUÇÃO

O processo de transformação chuva-vazão é de grande interesse em projetos ambientais e de engenharia, em gerenciamento de recursos hídricos e no manejo da bacia hidrográfica. Em geral este processo não é satisfatoriamente modelado por métodos simples, devido a sua natureza não linear e ao fato de que as variáveis envolvidas apresentam heterogeneidades temporais e espaciais [16].

Modelos chuva-vazão descrevem a relação entre a chuva e a vazão de uma bacia hidrográfica, quantificando a transformação da chuva em vazão de um curso de água. Esta relação é descrita através de um hidrograma, que mostra a variação da vazão do curso de água no tempo, de determinado ponto de interesse da bacia. Portanto, os modelos chuva-vazão têm diversas finalidades práticas para a previsão de cheias, da recarga de reservatórios, da disponibilidade de água para abastecimento urbano ou irrigação, dentre outras.

Diversos métodos de análise têm sido desenvolvidos com o objetivo de estimar a vazão com base na quantidade de chuva incidente sobre a bacia. Entretanto a estratégia utilizada por estes métodos, consiste na maioria das vezes, na simplificação dos fenômenos hidrológicos, sacrificando desse modo a fidelidade à realidade. Por outro lado, se o modelo assume um grande número de variáveis de entrada, isso adiciona uma complexidade não apenas computacional, mas também de modelagem do próprio problema.

Uma abordagem que surgiu nas últimas décadas, foi utilizar redes neurais para a modelagem de processos hidrológicos, em essência devido a duas particularidades das redes neurais: capacidade de resolução de problemas não-lineares e capacidade de aprendizagem.

Existem diversos interesses em desenvolver um modelo hidrológico de uma bacia, entre os quais pode-se citar [35] [32]: 1. previsão de enchentes; 2. melhor planejamento da agricultura; 3. previsão de nível de um reservatório; 4. qualidade da água; 5. previsão da vazão; 6. previsão de precipitação; 7. modelagem de águas subterrâneas; 8. previsão de secas; 9. estudo da relação chuva-vazão; 10. previsão de evaporação; 11. caracterização da poluição do solo.

Os dados de entrada para a modelagem da chuva-vazão são obtidos através de amostragem por séries temporais, sendo estes dados submetidos a

testes em uma rede neural, para a avaliação das características hidrológicas e previsões de vazão e chuva num momento posterior, através da simulação, na bacia hidrográfica. O desempenho de uma rede neural artificial pode ser avaliado através da exatidão de suas estimativas em comparação com resultados reais.

A abordagem mais encontrada na literatura sobre modelagem chuva-vazão e RNAs, foi aquela utilizando uma rede neural do tipo feedforward multicamadas, com aprendizagem supervisionada [32]. Doravante neste texto, será utilizado o termo redes neurais como sinônimo de redes neurais artificiais, e o modelo rainfall-runoff é traduzido aqui como modelo chuva-vazão.

O problema da modelagem pode se tornar realmente complexo quando o número de variáveis exigidas pelo modelo é grande ou variável, além do problema adicional, de que nem todas as bacias, no Brasil, dispõem de grande número de variáveis observadas. As redes neurais simplificam boa parte deste problema, pois conseguem representar razoavelmente bem o problema da transformação chuva-vazão com um conjunto menor de variáveis, levando em consideração o registro histórico das séries temporais, ou seja, o passado da bacia hidrográfica. Este tipo de modelo pode ser extremamente importante para regiões onde dados hidrológicos e climáticos são limitados [17].

Outro ponto importante é a natureza do processo de transformação de chuva sobre a bacia em vazão do rio. Esta é uma relação tipicamente não-linear. Por esta natureza, as redes neurais podem ser utilizadas com bastante sucesso, devido a sua característica para mapeamento de problemas lineares e não-lineares.

Neste trabalho é desenvolvida a modelagem da relação chuva-vazão da bacia do Rio Piquiri, cuja nascente se encontra no Município de Campina do Simão, Paraná, e cuja foz encontra-se junto ao rio Paraná.

1.1 OBJETIVOS

O objetivo deste trabalho é avaliar o uso das redes neurais artificiais para a modelagem hidrológica da transformação chuva-vazão em um ponto intermediário da bacia hidrográfica do Rio Piquiri, localizada no estado do Paraná (Brasil), e cuja foz encontra-se junto ao rio Paraná. Para isso foi utilizada uma base

de dados, de séries temporais, contendo informações de chuva-vazão para verificar a fidelidade dos dados previstos pela rede neural, com os dados observados.

Neste estudo é proposto um modelo híbrido a partir do acoplamento de um algoritmo genético juntamente com uma rede neural artificial. O objetivo do algoritmo genético é realizar uma busca global no espaço de estados do problema de modo a encontrar uma matriz de pesos que minimize uma função objetivo. Essa abordagem é interessante principalmente quando comparada com uma atribuição aleatória dos valores iniciais dos pesos, que nesse caso podem levar a um mínimo local no qual o algoritmo backpropagation tende a estagnar.

1.2 OBJETIVOS ESPECÍFICOS

Dentre os objetivos específicos deste trabalho pode-se citar os seguintes:

- Treinar uma RNA para modelar a chuva-vazão na bacia do Rio Piquiri.
- Validar a rede neural através de métodos estatísticos.
- Prever a vazão da bacia hidrográfica a partir de uma RNA.
- Selecionar um modelo de rede neural apropriado para as características da bacia hidrográfica.
- Comparar o desempenho da RNA com outras técnicas como por exemplo o modelo ARMA.
- Testar um modelo híbrido, usando algoritmos genéticos com redes neurais artificiais.
- Verificar qual técnica apresenta melhor resultado para se prever valores futuros da vazão, de modo que se possa ter uma gestão mais eficiente dos recursos hídricos de uma bacia hidrográfica.
- Investigar a adequação do uso de redes neurais artificiais para modelar a relação chuva-vazão em bacias hidrográficas.
- Identificação das configurações e parâmetros ideais de RNAs para a modelagem da chuva-vazão. Investiga-se nesse sentido diversas modalidades de redes a fim de selecionar a melhor opção.

1.3 JUSTIFICATIVA

Diversas modalidades de estudo de modelagem da vazão em uma bacia já foram propostas. Dentre elas, há as que utilizam além da precipitação, a temperatura e a evapotranspiração. Neste estudo, optou-se pela simplificação do número de variáveis do modelo, que parte apenas da precipitação e da vazão como fonte de dados para treinamento da RNA. Esta abordagem visa abarcar um número maior de bacias hidrográficas, observando-se que nem todas possuem uma ampla ou variada coleta de variáveis hidrometeorológicas. Desse modo, pode-se ter um modelo com maior abrangência de aplicabilidade e principalmente incluir na modelagem aquelas bacias com monitoramento limitado. A abordagem apresentada neste trabalho, busca simplificar ao máximo a representação do fenômeno de transformação chuva-vazão, no momento em que as variáveis consideradas nesta modelagem restringem-se a precipitação e vazão apenas, como justificativa principal na modelagem da vazão. Entretanto este não é um fenômeno simples, conforme apresentado em [49], onde modelagens tradicionais podem considerar inúmeras variáveis.

Desse modo uma das principais justificativas deste trabalho está no fato de que uma bacia hidrográfica adequadamente monitorada não é uma realidade nas bacias hidrográficas brasileiras em geral. Muitas bacias possuem períodos de ausência de monitoramento, limitação em termos de equipamentos de monitoramento, falta de recursos em ampliação na malha de estações hidrometeorológicas. Em resumo, a aplicação de modelos mais sofisticados podem ser inviáveis, dependendo do estado em que se encontra o monitoramento de uma bacia. Desse modo, a metodologia apresentada aqui, visa abordar um processo no qual busca-se a simplificação para aumentar-se a abrangência de modelagem de um número maior de bacias hidrográficas. A contribuição dessa pesquisa para o cenário hidrológico é justamente fornecer uma metodologia que permite um alcance de modelagem maior em cenários onde a existência de informações é restringida.

No cenário computacional, pretende-se comparar diferentes métodos computacionais avaliando-se em comum, suas capacidades de ajustes ao conjunto de dados para realizar previsões.

1.4 ESTRUTURA DO TRABALHO

A seguir apresenta-se os capítulos deste trabalho estruturados em três partes: (1) Fundamentação teórica, (2) Metodologia e (3) Resultados e Discussões.

1.4.1 Fundamentação Teórica

São apresentados os principais conceitos da Hidrologia com relação as características e variáveis relacionadas a questão da vazão e da chuva numa bacia hidrográfica. Conceitos de área de drenagem, escoamentos, ciclo hidrológico e bacias hidrográficas são explanados.

Em termos de processamento dos dados coletados das estações meteorológicas da bacia hidrográfica, são apresentados conceitos importantes dentro da análise de séries temporais, utilizados principalmente pelos métodos autorregressivos e no pré-processamento dos dados, como médias móveis, filtros e autocorrelação.

Para conceitualização da técnica aplicada neste trabalho é realizado um estudo teórico dos modelos de redes neurais artificiais, suas principais características, estrutura e parâmetros decisivos de configuração.

Discute-se ao final, sobre os algoritmos genéticos, paradigma evolucionista aplicado para refinar e melhorar a convergência das redes neurais.

1.4.2 Metodologia

Apresenta-se neste capítulo a bacia hidrográfica do Rio Piquiri, utilizada como estudo de caso. Além disso é mostrado as etapas de pré-processamento dos dados e a seleção das estações meteorológicas relevantes para a área de estudo. Ao fim do capítulo são propostos os modelos de entrada de dados

que serão utilizados para a definição das redes neurais que serão testadas para o experimento.

1.4.3 Resultados e Discussão

Neste capítulo são apresentados os resultados das simulações utilizando as redes neurais de acordo com os modelos propostos no capítulo anterior. Busca-se nesse caso avaliar a capacidade da rede neural na modelagem do processo de transformação da chuva em vazão. Para isso os dados obtidos da série temporal, equivalente a um período de 9 anos, são divididos em dois conjuntos, um para treinamento e validação e outro para teste. Os dados de validação são utilizados para seleção do melhor modelo de acordo com os critérios de menor erro (RMSE) e do indicador de Nash-Sutcliffe.

Também neste capítulo, o mesmo conjunto de dados utilizado no treinamento e aprendizagem da RNA é utilizado para a simulação do mesmo processo chuva-vazão, porém usando a metodologia box e jenkins - através de processos estocásticos denominados ARMA (autorregressive moving average), para a previsão de séries temporais. Ao final compara-se os resultados obtidos pelos modelos propostos, verificando-se qual a metodologia mais adequada para a representação do processo de transformação chuva-vazão.

1.5 REVISÃO BIBLIOGRÁFICA

Diversos estudos já foram realizados com o uso de redes neurais artificiais aplicadas na área da modelagem dos fenômenos hidrológicos. As RNAs são particularmente interessantes para essa modelagem, devido a natureza não-linear e complexa desses fenômenos. A seguir é apresentado em ordem cronológica alguns desses estudos e seus resultados, em especial aqueles direcionados a modelagem do processo de transformação chuva-vazão.

Minns [22] et al. (1996), propõem um estudo onde dados sintéticos

de vazão são gerados para testar a eficiência de uma RNA em modelar a relação chuva-vazão. Propõem 4 modelos de redes neurais, variando o lag de entrada e o número de neurônios na camada oculta e o número de camadas intermediárias. Concluem que as RNAs são capazes de identificar a relação entre vazão e precipitação, onde nos testes propostos, a rede com 4 camadas apresentou pouca melhora em relação a rede com 3 camadas. Sugerem ao fim que aumentando o número de neurônios da camada oculta para um número entre 10 a 15 neurônios é suficiente para eliminar a necessidade de uma quarta camada na rede.

Shamseldin [55] (1997) conduz um estudo onde usa uma RNA para modelar a relação chuva-vazão em 6 diferentes bacias hidrográficas, usando dados de medições diárias. Para efeitos comparativos, os resultados obtidos com a RNA são comparados com três outros modelos: SLM (simple linear model), LPM (linear perturbation model) e>NNLPM (Nearest neighbour linear perturbation model). Ao final conclui que a RNA apresenta resultados variados, ou seja, apresentou boa performance em algumas bacias, e fraco desempenho em outras bacias.

Dawson [10] et al. (1998) propõem o uso de redes neurais para a previsão da vazão e cheias nos rios Amber e Mole no Reino Unido com área de drenagem de aproximadamente 140km^2 . O modelo proposto trabalha com dados de 15 minutos, prevendo cheias, num horizonte de até 6 horas. Para efeitos comparativos a RNA foi comparada com o sistema de previsão de cheias (Flood Forecasting System - FFS) da agência Severn-Trent Environment. Os resultados mostraram uma capacidade maior de margem de horizonte de previsão da RNA em relação ao FFS (de até 2 horas). Além disso a RNA foi capaz de construir um modelo robusto com um conjunto pequeno de dados de calibração, ao todo de 100 dias, utilizando o período do outono e início do inverno.

Campolo [6] (1999) et al. utilizam uma rede neural para analisar e prever o comportamento do rio Tagliamento na Itália, durante períodos de fortes chuvas. O modelo proposto faz uso de diversas estações de pluviometria levando em consideração sua distribuição espacial. A área de drenagem da bacia tem um total de 2480km^2 incluindo diversas sub-bacias. No estudo foram propostos dois modelos de previsão do nível de água: o primeiro modelo para curtos períodos de tempo (1 hora); o segundo modelo para tempos longos acima de 10 horas. O coeficiente de R^2 do modelo de RNA proposto pelos autores foi de 98% tanto para

calibração quanto para validação.

Tokar e Johnson [58] (1999) aplicam redes neurais para realizar previsões de vazão diária, como uma função da precipitação, temperatura e derretimento da neve medidos diariamente, na bacia do rio Little Patuxent em Maryland, Estados Unidos. A RNA é comparada a técnicas de regressão estatística e com o modelo SCRR (simple conceptual rainfall-runoff), proposto com McCuen e Snyder. Concluem ao final que o modelo obtido com a RNA mostrou resultados mais sistemáticos, com necessidade de menos dados para calibração e maior precisão na previsão. Também é apresentado o impacto na eficiência da previsão da RNA de acordo com a seleção dos dados de treinamento. A arquitetura da melhor rede obtida na modelagem apresentou 3 camadas, com 10 neurônios na camada oculta, sendo que a saída da rede (vazão - $Q(t)$) ficou em função de 3 entradas, $Q(t)=f\{P(t),P(t-1),T(t)\}$, onde P representa a precipitação e T representa a temperatura.

Santos [16] (2001) realiza um estudo onde aplica uma RNA para o diagnóstico e prognóstico de vazão em bacias urbanas da Região Metropolitana de São Paulo. Utiliza dados do radar meteorológico e dados telemétricos da bacia do Alto Tietê. A RNA usada é do tipo MLP com algoritmo de treinamento LLSSIM (Linear Least Square SIMplex). Para comparação dos resultados, utiliza um modelo auto-regressivo. Conclui ao final, que as RNAs apresentam resultados superiores, sendo confiáveis num intervalo de até 90 minutos a frente.

Lallahem [20] (2003) et al. utilizaram redes neurais artificiais do tipo MLP treinada com backpropagation para simular a vazão de águas subterrâneas na região norte da França, localizada nos limites da bacia de Paris, com chuva média anual de 450 e 1000 mm, e área de 18km^2 . Como entrada da rede utilizaram diversas variáveis: nível d'água, precipitação, temperatura, vazão, evapotranspiração, infiltração e vazão na fissura. Os dados coletados compreendiam o período de 1 ano, entre janeiro a dezembro de 2000, em escala diária. Desse total utilizaram 50% para treinamento, 25% para teste e os outros 25% para validação. Os resultados mostraram um coeficiente R^2 de 98%.

Rajurkar [31] et al. (2003) apresentam uma abordagem para modelagem de vazões diárias durante eventos de enchentes utilizando RNAs. A modelagem chuva-vazão foi realizada, acoplando um modelo SLM (simple linear

model) auxiliar a rede neural. Este modelo SLM tem por objetivo representar o estado da mistura do solo, de modo que essa informação seja utilizada juntamente com a precipitação como entrada na rede neural. Esse argumento deve-se a estudos anteriores, no qual mostrou-se que modelos de RNAs que levavam em consideração apenas a precipitação como entrada, apresentaram baixa capacidade de representação da relação chuva-vazão. Entretanto quando usados em conjunto com a vazão anterior, o modelo apresentou melhoras significativas. Desse modo o uso do modelo SLM vem a contribuir nesse sentido, oferecendo um parâmetro (a vazão estimada anterior) de entrada adicional para melhorar a previsão da mesma. A metodologia proposta foi aplicada em 5 diferentes bacias hidrográficas, localizadas em diferentes regiões geográficas, com resultados satisfatórios, e desse modo os autores mostraram a capacidade versátil das RNAs em lidar com cenários adversos.

S.Riad [32] (2004) et al. utilizaram uma RNA para realizar a modelagem da relação chuva-vazão numa bacia hidrográfica localizada na região semi-árida no Marrocos. Para esse propósito foi usada uma rede do tipo MLP, treinada com o algoritmo backpropagation, utilizando dados de um período de 7 anos, com medições diárias da precipitação e vazão. Ao final a RNA proposta é comparada com um modelo de regressão, onde ficou evidenciado que as RNAs conseguiram obter um modelo de representação mais acurado.

Machado [48] (2005) investiga o processo de modelagem chuva-vazão através do uso de uma rede neural do tipo MLP. A modelagem foi realizada na bacia hidrográfica do rio Jangada, localizado no estado do Paraná, Brasil. Para validação dos resultados obtidos pela RNA o autor propõe um comparativo com o modelo conceitual IPHMEN que é um modelo desenvolvido pelo Instituto de pesquisas hidráulicas da UFRGS [15]. Utilizando diversas configurações de quantidade de neurônios na camada oculta e parâmetros de entrada, demonstrou ao final que o modelo de rede no qual considera precipitação, evapotranspiração e vazão como entradas, apresentaram o melhor resultado. No confronto com o modelo IPHMEN as redes neurais apresentaram um melhor desempenho.

Nasseri [51] (2008) et al. utilizam redes neurais artificiais para a previsão de precipitação na bacia de Upper Parramatta localizada em Sydney, Austrália, com área aproximada de 112km^2 . Os autores relatam a problemática e o consumo de tempo para encontrar e selecionar uma arquitetura de rede neural

adequada. Para contornar tal questão, os autores propõem a utilização de algoritmos genéticos acoplado a uma rede neural tipo MLP, para melhorar o desempenho da rede durante a fase de treinamento. O uso de algoritmos genéticos é voltado nesse trabalho para determinar os pesos correspondentes das conexões entre cada neurônio artificial. Os resultados obtidos indicaram uma melhora e redução no erro da rede, quando comparado a modelagem utilizando apenas a MLP com backpropagation.

Sedki [54] et al. (2009) investigam o uso de algoritmos genéticos acoplados a redes neurais para prever a vazão de uma bacia hidrográfica. Utilizam a precipitação e a vazão como variáveis de entrada da rede. Usam codificação por valor para a codificação dos cromossomos. O vetor entrada da rede utilizado continha os valores de precipitação e vazão de até 4 dias anteriores ($t-4$), sendo que a melhor rede encontrada pelos autores foi aquela com (4-5-1). Nessa metodologia, o AG executava por 80 gerações, seguido de um treinamento usando backpropagation. O resultado final mostrou um coeficiente de eficiência (R^2) de 0,91.

No quadro 1.1 é apresentado uma síntese da metodologia desenvolvida nos trabalhos discutidos anteriormente, com os principais parâmetros e técnicas aplicadas.

1.6 SOBRE ESTE TRABALHO

Esta pesquisa realiza um estudo comparativo entre 3 diferentes metodologias para a modelagem do processo chuva-vazão, onde avalia-se a capacidade de previsão de cada modelo. Em geral, nos trabalhos apresentados, é feita uma comparação entre duas metodologias numa única escala temporal (tipicamente em escala diária). Avalia-se aqui o desempenho dos modelos em duas escalas temporais: diária e mensal, e 3 metodologias distintas, nesse caso, modelos usando redes neurais do tipo MLP, redes neurais MLP com algoritmos genéticos e modelos autorregressivos com médias móveis.

Autor	País	Bacia	Área	Entradas	Saída	Escala	Método	Arquitetura
Riad [32]	Marrocos	Ourika	503 km ²	P, Q	Q	Diário	MLP com BP	14-12-1
Campolo [6]	Itália	Tagliamento	1950 km ²	P, N	N	Horas	MLP com BP	8-3-1
Dawson [10]	Inglaterra	River Amber	139 km ²	P, Q, S, PMA, QMA	Q	Minutos	MLP com BP	15-10-1
Lallahem [20]	França	Paris	18 km ²	P, T, E, N, QF, QS	Q	Diário	MLP com BP	6-1-1
Nasseri [51]	Austrália	Upper Parramatta	112 km ²	P	Q	Minutos	AG e MLP com BP	***
Shamseldin [55]	***	6 bacias	***	P	Q	Diário	MLP com gradiente conjugado	5-4-1
Rajurkar [31]	vários	7 bacias	***	P, Q	Q	Diário	SLM e MLP com BP	várias
Tokar [58]	EUA	Little Patuxent	98 km ²	P, T	Q	Diário	MLP com BP	3-10-1
Machado [48]	Brasil, Paraná	Jangada	1055 km ²	P, E, Q	Q	Diário	MLP com Levenberg	5-4-1
Santos [16]	Brasil, São Paulo	Alto Tietê	310 Km ²	P, Q	Q	Minutos	MLP com LLSSIM	9-1-1
Minns [22]	***	dados sintéticos	30 km ²	P, Q	Q	Horas	MLP com BP	17-8-1
Sedki [54]	Marrocos	Ourika	503 km ²	P, Q	Q	Diário	AG e MLP com BP	4-5-1

Quadro 1.1 - Estudos apresentados e respectivas configurações utilizadas ¹

Além disso, o objeto de estudo, a bacia hidrográfica do rio Piquiri, possui área de drenagem consideravelmente maior que a média dos trabalhos revisados. A bacia hidrográfica em questão tem aproximadamente 24.731km² de área total, sendo que utilizou-se uma região de estudo com área de 3.620km². Isto acrescenta uma complexidade maior no momento em que a variabilidade espacial e temporal exercem maior influência na correlação entre os dados de precipitação e os dados de vazão. Isso significa dizer que quanto maior a área de drenagem de uma bacia hidrográfica, maiores são os fatores ou variáveis de influência sobre o escoamento de um canal.

Em termos hidrológicos, tem-se a definição de uma técnica de modelagem que reduz a duas variáveis ou parâmetros necessário para a calibração e especificação de um modelo, apresentando uma evidente vantagem quando comparado a outras técnicas tradicionais que exigem diversas variáveis para calibração do modelo.

¹ BP - *backpropagation* (gradiente descendente); MLP - multilayer perceptron; AG – algoritmo genético; P - precipitação; Q - Vazão; T - temperatura; E - evapotranspiração; N - nível da água; QF - vazão de infiltração ; QS - vazão da fissura; PMA - média móvel da precipitação; QMA - média móvel da vazão; S - tempestades

2 FUNDAMENTOS TEÓRICOS

Neste capítulo são apresentados os principais conceitos sobre hidrologia, análise de séries temporais e modelos autorregressivos, redes neurais artificiais e algoritmos genéticos.

2.1 MODELOS HIDROLÓGICOS

Nesta seção apresenta-se e caracteriza-se brevemente conceitos fundamentais dentro do entendimento dos princípios da modelagem do processo de transformação chuva-vazão, em particular sob a ótica da hidrologia. O objetivo de estudo dessa ciência diz respeito principalmente a distribuição e movimento da água tanto acima quanto abaixo da superfície da terra e também na atmosfera [9].

2.1.1 Ciclo hidrológico

O ciclo hidrológico (figura 2.1), ou ciclo d'água, é um fenômeno de escala global que descreve o movimento da água entre a superfície terrestre e a atmosfera, estimulado pela energia solar [4]. A água pode assumir diversos estados, entre os quais: líquido, vapor e gelo, em vários momentos dentro do ciclo hidrológico. O calor provocado pelo sol atinge e aquece a água no oceano, rios e lagos, ocasionando a evaporação para o ar. Gelo e neve também podem transformar-se diretamente em vapor d'água. Evapotranspiração é a transpiração da água a partir das plantas e da evaporação a partir do solo. Ventos ascendentes levam o vapor d'água para a atmosfera onde em altitudes mais altas, com temperaturas mais baixas, levam a condensação do vapor em nuvens. Correntes de ar movem as nuvens ao redor do globo, onde eventualmente as partículas de água colidem e caem de acordo com a força da gravidade como precipitação. A

precipitação pode ser de neve, granizo ou gotículas de água. Desse modo a água retorna aos oceanos, lagos, rios e também aos solos através da infiltração.

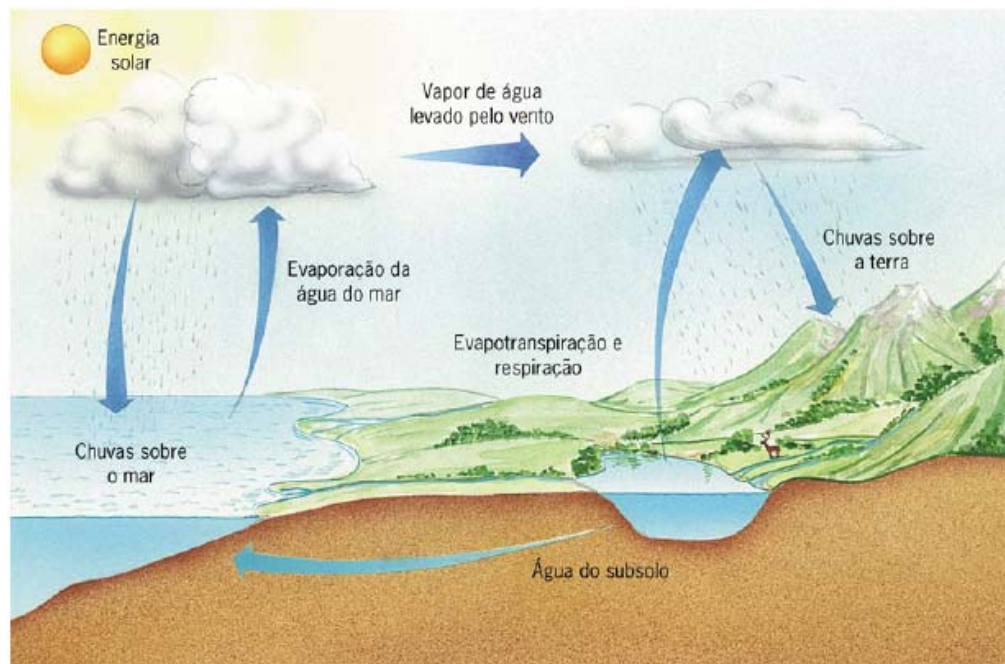


Figura 2.1 – Ciclo hidrológico [2]

O ciclo pode ser dividido nas seguintes etapas:

- **Precipitação:** consiste no vapor de água condensado que cai sobre a superfície terrestre.
- **Infiltração:** consiste no fluxo de água da superfície que se infiltra no solo.
- **Escoamento superficial:** é o escoamento das águas na superfície terrestre para os mares.
- **Evaporação:** é a transformação da água no seu estado líquido para o estado gasoso à medida que se desloca da superfície para a atmosfera.
- **Transpiração:** é a forma como a água existente nos organismos passa para a atmosfera.
- **Evapotranspiração:** é o processo conjunto pelo qual a água que cai é absorvida pelas plantas, voltando à atmosfera através da transpiração ou evaporação direta (quando não absorvida).
- **Condensação:** é a transformação do vapor de água em água líquida, com a criação de nuvens e nevoeiro.

2.1.2 Bacia hidrográfica

Bacias hidrográficas (figura 2.2) são regiões ou áreas rodeadas por cumes de montanhas, formadas por desníveis na geografia do terreno, onde a água proveniente da precipitação é drenada para um curso de água e seus afluentes, seja um córrego, lago ou rio. Estas regiões são limitadas por divisores de águas que são em si cumes de montanhas, que forçam a água a seguir determinado caminho e muitas vezes separam geograficamente a água proveniente da chuva para cursos de água (rios) distintos. Essas separações, que orientam a água para diferentes corpos d'água, são geralmente utilizadas como critérios de demarcação entre diferentes bacias hidrográficas, ou para indicar onde começa e termina a bacia hidrográfica. Sob outro ponto de vista, pode-se definir uma bacia hidrográfica como a área de captação de água, ou a área de drenagem de água para um rio principal ou afluente.

No estudo da hidrologia, a bacia hidrográfica é considerada a unidade espacial mais comum. Pode ser considerada como a área de terra na qual a água escoar para um rio, e o rio escoar para o oceano. Isso sugere que numa bacia, toda a água incidente na mesma escoar para um ponto central (de menor altitude). A característica comum a qualquer ponto de uma bacia é que independentemente de onde a água da chuva incida, ela irá terminar sempre num mesmo local: onde o rio encontra o mar, mesmo que isso seja de uma forma indireta, por exemplo, quando uma bacia está contida em outra maior [5]. O tamanho de uma bacia hidrográfica pode variar desde alguns hectares até milhares de quilômetros quadrados.

Uma bacia hidrográfica pode ser definida em termos de sua topografia, assumindo-se que toda água que incide em sua superfície escoar para baixo. O pressuposto de que toda água escoar para um rio não é sempre verdadeira, especialmente quando a geologia da bacia é complexa. É possível que a água possa fluir como água subterrânea para outra bacia, criando um problema de como definir a área da bacia. Apesar de tais questões, a bacia hidrográfica fornece uma importante unidade espacial para os hidrólogistas no intuito de estudar como a água é distribuída e movimenta-se em certos períodos de tempo.

Área de drenagem da bacia é uma extensão de terra onde a água da chuva ou do derretimento da neve é drenada pela força da gravidade para uma

região mais baixa, até confluir em um corpo d'água, tal como um rio, lago, reservatório, estuário e oceano. A área de drenagem da bacia age como um funil, coletando toda a água dentro de sua área e canalizando-a dentro de um fluxo maior de água. Cada drenagem da bacia é separada topograficamente por outros limites de bacias, tais como barreiras como montanhas e cordilheiras, que são conhecidas como divisores de água.



Figura 2.2 – Bacia Hidrográfica - [45]

2.1.3 Precipitação

Precipitação é o processo de liberação de água da atmosfera para a superfície da terra. O termo abrange todas as formas de liberação de água, incluindo: neve, granizo e chuva. Consiste na principal fonte de entrada de água numa bacia hidrográfica. A chuva é um fenômeno relativamente simples de ser medido, entretanto difícil de ser mensurado precisamente, sendo extremamente variável no tempo e no espaço.

Em termos de análise hidrológica é importante conhecer a quantidade de precipitação. O método tradicional é através da medida da quantidade de água líquida vertical num recipiente. É medida em milímetros ou centímetros em profundidade. A medição é portanto a altura da água que acumularia na superfície se toda a chuva que ali incidiu permanecesse. Devido ao fato de haver uma grande variação da medição da precipitação em uma pequena área da bacia, pode-se

constatar que de fato todas as medições de precipitação são na verdade estimativas do volume de chuva incidente na bacia.

O instrumento utilizado para medir a chuva é chamado pluviômetro. Este instrumento mede a altura da chuva de distribuição horizontal supostamente homogênea e não submetida à evaporação [11].

De modo geral a precipitação incidente numa bacia hidrográfica que atinge a superfície do solo poderá: evaporar, infiltrar no solo ou escoar superficialmente [4].

2.1.4 Evaporação

Evaporação é a transferência de água líquida em um estado gasoso e a sua difusão na atmosfera. Para que isso ocorra deve haver água em estado líquido e energia suficiente do sol ou da atmosfera. A contribuição da evaporação no ciclo hidrológico depende em grande parte da quantidade de água e de energia disponível, ou seja, dois fatores determinados pelo clima da região. Durante os meses de inverno em climas úmidos-temperados, evaporação é um componente de menor representação no ciclo hidrológico, pois há pouca energia disponível exercendo influência no processo de evaporação. Isso muda durante o verão quando a energia passa a ser abundante e a evaporação assume um papel principal no ciclo da água. Também pode-se encontrar esse cenário em climas extremamente quentes e áridos onde há geralmente grande quantidade de energia para a evaporação, mas pouca água para ser evaporada. Como consequência a quantidade de evaporação é pequena.

2.1.5 Vazão

A quantidade de água dentro de um rio ou fluxo representa o produto final de todos os outros processos do ciclo hidrológico, sendo desse modo um dos principais fatores na análise hidrológica. Vazão é um termo amplo que abarca o

movimento da água para um fluxo canalizado depois de ter atingido a superfície como precipitação [9]. Esse movimento pode ocorrer tanto abaixo como acima da superfície em diferentes velocidades. Uma vez que a água alcança um fluxo ela move-se em direção ao oceano de uma forma canalizada, e é chamada como escoamento fluvial. Esse escoamento é medido como o volume de água em um determinado período (descarga).

A vazão é medida em metros cúbicos por segundo (m^3/s), ou seja, indica a velocidade com a qual um volume escoar. O registro contínuo da vazão é chamado de hidrograma. Tipicamente esta medida é uma média da vazão de um período de tempo sobre uma série de amostras (ex.: minutos, horas de registros). Os picos encontrados num hidrograma (figura 2.3) referem-se a quantidade de água no rio, durante e imediatamente depois de uma chuva significativa.

A saída (outlet) é o ponto mais baixo da bacia hidrográfica, tipicamente dentro de um canal de escoamento, e geralmente junto a uma união com um corpo d'água.

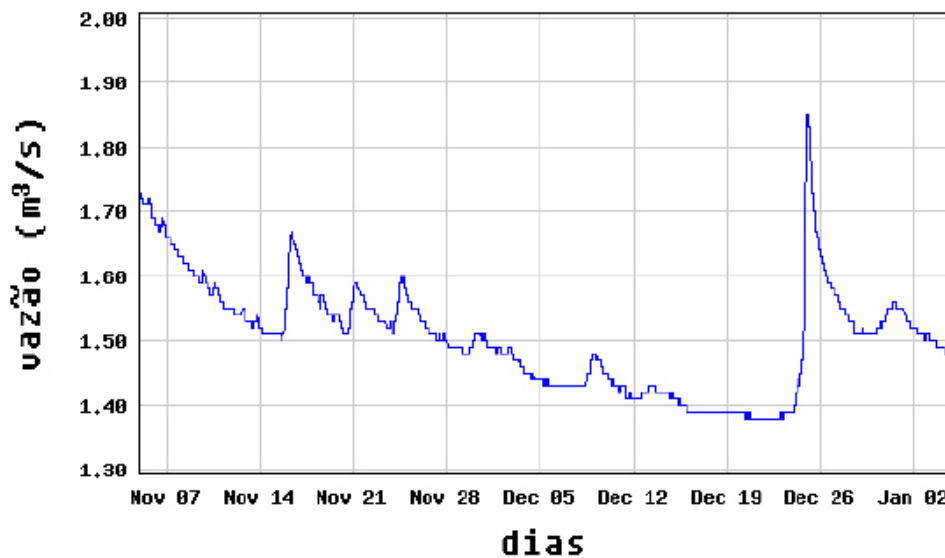


Figura 2.3 – Hidrograma com média diárias, do rio Baron Fork, AR, EUA [56]

Modelo chuva-vazão

A vazão num rio ou canal é um processo complexo influenciado por fatores como a topografia, a cobertura vegetal, tipos de solos, características do

canal, presença de aquíferos, distribuição da precipitação e área urbanizada [4].

O modelo chuva-vazão parte de uma abordagem mais simplificada dos processos do ciclo hidrológico, tendo como objetivo principal avaliar qual a vazão de uma bacia hidrográfica em função de uma precipitação qualquer. Estes modelos descrevem em função da chuva, as perdas por evaporação, interceptação, infiltração e percolação da água subterrânea [48].

Na hidrologia um dos principais modelos chuva-vazão é o SSARR(*Streamflow Synthesis and Reservoir Regulation*), proposto pela U.S Army Corps Engineers em 1958. Em 1966 Crawford e Linsley apresentam o Standford IV, com discretização a cada 15 minutos, sendo considerado um dos mais completos em relação a quantidade de processos representados [48].

2.2 ANÁLISE DE SÉRIES TEMPORAIS

Nesta seção apresenta-se o conceito de série temporal e as principais técnicas de análise de séries temporais, tais como as médias móveis, filtros e os métodos autorregressivos.

2.2.1 Introdução

Uma série temporal consiste tipicamente de uma sequência de dados, medidos sucessivamente no tempo. Análise de séries temporais abrange a aplicação de métodos utilizados para compreender a natureza da série temporal, tais como, a função que gerou a série, de onde os dados vieram, como os dados foram obtidos e qual a dependência e a correlação das observações vizinhas. Além disso tem-se os métodos de previsão, ou seja, a partir de eventos passados, aqueles registrados pela série, prever eventos futuros, ainda desconhecidos.

Pode-se considerar uma série temporal como um sistema dinâmico cuja caracterização matemática é desconhecida, ou seja, tudo o que se conhece

sobre o sistema é um conjunto de observações gerados pelo sistema em instantes de tempo discretos, tipicamente a uma taxa uniforme.

A análise de séries temporais tem um papel fundamental em diversos campos da ciência, tais como nas ciências sociais, preocupadas com números e taxas em relação ao crescimento populacional e seu desenvolvimento em períodos de tempos; nas ciências biológicas interessadas na evolução do número de casos de infecção de alguma doença; e nas ciências econômicas, procurando por padrões recorrentes nas altas e baixas de preços das commodities por exemplo. Além desses campos, as séries temporais são utilizadas amplamente nas análises climáticas em especial na previsão de chuva e secas e na qualidade da água.

Segundo [27], antes de se compreender os métodos estatísticos para análise de séries temporais, é importante identificar o domínio no qual se fará a análise das séries temporais, tipicamente identificados em duas categorias: 1. abordagem por domínio de tempo (*time domain approach*); 2. abordagem por domínio de frequência (*frequency domain approach*). Importante ressaltar que ambos domínios não são, necessariamente, mutuamente exclusivos.

A abordagem por domínio de tempo é geralmente motivada pela pressuposição de que a correlação entre pontos adjacentes no tempo é melhor explicado na dependência da relação dos valores atuais nos valores passados. Esta abordagem se atém na modelagem de valores futuros de uma série temporal como uma função paramétrica dos valores atuais e passados. Esta modelagem possibilita usar os resultados como ferramenta de previsão.

A abordagem por domínio de frequência assume que a característica de interesse principal está relacionada com a variação sinusoidal periódica ou sistemática encontrada naturalmente na maioria dos dados. Estas variações periódicas são geralmente causadas por fenômenos biológicos, físicos ou ambientais. Por exemplo: a vibração do vento que afeta a asa de um avião; a temperatura da superfície do mar causada pelo El Niño, afetando o número de peixes no oceano [27].

O principal objetivo da análise de séries temporais é desenvolver um modelo matemático que forneça uma descrição plausível para um conjunto de dados e assim adquirir conhecimento e informações sobre padrões implícitos que desenvolvem-se ao longo de um intervalo de tempo, tipicamente através da análise e observações de dados passados [13].

O passo seguinte após a construção do modelo, consiste na extrapolação do modelo para além do domínio temporal conhecido, obtendo assim a previsão do modelo. A definição e extrapolação do modelo é convencionalmente denominado por análise de séries temporais [13].

A série será denominada contínua se o conjunto de dados observados for contínuo, sendo a série denotada por $\{X(t) : t \in T\}$, onde T é um conjunto $T = \{t : t_1 < t < t_2\}$.

Caso o conjunto seja discreto a série é chamada discreta, ou seja, quando as observações são obtidas em tempos específicos, geralmente em intervalos de tempo iguais.

A variável independente ou o intervalo de tempo é denominado: t_1, t_2, \dots, t_n .

A variável dependente ou os dados observados (em função do tempo) são denotados: $y(t_1), y(t_2), \dots, y(t_n)$ ou simplesmente y_1, y_2, \dots, y_n .

De modo geral, os principais objetivos do estudo das séries temporais podem ser classificados do seguinte modo [7]:

1. **Analisar.** Analisar e descrever o padrão de tendência da série, a identificação da influência do efeito da sazonalidade ou de alguma variação cíclica e anomalias (observações fora do padrão).
2. **Explicar.** Quando a observação é realizada em duas ou mais variáveis, é possível a utilização da análise da variação de uma série temporal para explicar a variação de outra série. Isto pode levar a um entendimento maior do mecanismo que gerou a série. Os modelos de regressão múltipla podem ajudar nesta tarefa.
3. **Predizer.** Prever valores futuros a partir de valores passados (observados), dentro de uma incerteza, pois as previsões não são perfeitas.
4. **Controlar.** A utilização de séries temporais para o controle de qualidade de um processo.

Pode-se destacar 4 características ou componentes importantes das séries temporais [1], que influenciam seu comportamento:

1. **Tendência:** movimento direcional e contínuo ao longo de um período. De um modo geral, tendência pode ser definida por uma das componentes: uma tendência linear e um ruído associado, onde uma observação num tempo t é

uma variável X_t dada por: $X_t = \alpha + \beta_t + \epsilon_t$, sendo α e β constantes e ϵ um erro aleatório cuja média é zero.

2. **Sazonalidade:** oscilações de aumento ou diminuição que ocorrem sempre em um mesmo período de tempo. Em diversos tipos de observações podem ocorrer padrões de comportamento que tendem a se repetir de tempos em tempos.

3. **Movimentos cíclicos:** variações associadas a algum ciclo ou evento periódico, causando desvios em torno da tendência.

4. **Variações Irregulares:** movimento de natureza aleatória causada por algum evento imprevisto.

Uma característica especial da análise das séries temporais é o fato de que observações sucessivas não são, geralmente, independentes, e portanto a análise deve levar em consideração a ordem temporal em que as observações foram realizadas. Quando sucessivas observações são dependentes, valores futuros podem ser previstos de observações passadas. Se uma série temporal pode ser prevista exatamente, ela é chamada determinística. Entretanto a maioria das séries são estocásticas, ou seja, o futuro é apenas parcialmente determinado pelos valores passados, tornando impossível ter previsões exatas, devendo ser substituídas pela ideia de que valores futuros têm uma distribuição que é condicional ao conhecimento dos valores passados [7].

Portanto, as séries temporais podem ser classificadas quanto ao seu processo gerador de dados, em dois tipos [1]:

1. **Estocástico:** os dados observados surgem por meio de eventos aleatórios.

2. **Determinístico:** os dados observados não são regidos por um processo aleatório, comportando-se de acordo com alguma equação matemática, sendo a realização da predição, basicamente a aplicação da equação.

Quanto ao comportamento de uma série temporal pode-se dividir e analisá-la sobre o aspecto de 3 componentes:

$$X_t = T_t + S_t + R_t$$

onde T_t é a tendência, S_t é a componente sazonal e R_t representa o ruído aleatório. A componente S_t repete-se num intervalo fixo de tempo.

2.2.2 Métodos estatísticos

A seguir são apresentados alguns métodos estatísticos, usados em geral na análise das séries temporais. Cada método tem um foco específico na análise do comportamento de uma série histórica, desde médias simples, até os modelos autorregressivos.

2.2.3 Previsão

A proposta de um sistema de previsão é reduzir o risco na tomada de decisão. Previsões geralmente carregam erros, entretanto a magnitude deste erro dependerá do sistema de previsão utilizado e do critério ou margem de tolerância. Pode-se dizer que a tomada de uma decisão é a pressuposição de que a previsão está correta adicionando uma margem ou desconto de um possível erro da previsão (margem de segurança). Além disso, toda previsão exige o estabelecimento de um período e um horizonte de previsão. O período diz respeito a unidade base de tempo, no qual as previsões são realizadas (ex.: diário, mensal, anual, ...). Já o horizonte de previsão consiste no número de períodos futuros que serão alcançados pela previsão, como por exemplo, a previsão de gastos mensais para os próximos 12 meses (horizonte de 12 períodos).

2.2.4 Média Móvel (MA)

Tipicamente utilizada para analisar um conjunto de pontos, definindo para isso uma nova série das médias dos subconjuntos que compõe a série em análise. A média móvel é geralmente utilizada para suavizar ou reduzir flutuações locais e ressaltar tendências e ciclos em longos períodos de tempo.

Uma média móvel simples (SMA) é uma média que não possui pesos associados aos pontos ou dados considerados. Por exemplo, uma média

móvel da precipitação média diária, dos últimos sete dias pode ser definida pela equação (2.1):

$$SMA = \frac{p_n + p_{n-1} + \dots + p_{n-6}}{7} \quad (2.1)$$

A média móvel simples não é geralmente recomendada para medição da tendência, entretanto pode ser útil para remover variação de sazonalidade [7].

2.2.5 Filtros

São geralmente definidos para produzir uma saída com ênfase na variação de frequências específicas. Por exemplo, para obter dados mais suaves, pode-se remover a flutuação local que constitui o que se conhece por variação de alta-frequência. Em outras palavras, deseja-se um filtro de passa-baixa. Basicamente um filtro converte uma série X_t em outra Y_t através de uma operação linear [7]. Um exemplo de uma operação de filtragem pode ser ilustrado na seguinte fórmula (2):

$$y_t = \sum_{r=-q}^{+s} a_r x_{t+r} \quad (2)$$

onde a_r é um conjunto de pesos. Para atenuar as flutuações locais e estimar a média local, escolhe-se os pesos de modo que $\sum a_r = 1$, caracterizando tipicamente uma operação de média móvel. Um filtro útil para remoção de variação sazonal é o filtro de atenuação simétrica, que é uma média móvel simples onde o peso é definido por $a_r = 1/(2q + 1)$ para $r = -q, \dots, +q$, e o valor x_t atenuado é definido por (3):

$$x_t = \frac{1}{2q + 1} \sum_{r=-q}^{+q} x_{t+r} \quad (3)$$

2.2.6 Sazonalidade

Três modelos são usados para descrever sazonalidade:

$$A \quad X_t = m_t + S_t + \epsilon_t$$

$$B \quad X_t = m_t S_t + \epsilon_t$$

$$C \quad X_t = m_t S_t \epsilon_t$$

onde, m_t é o nível médio de-sazonalidade no tempo t , S_t é o efeito sazonal no tempo t e ϵ_t é o erro aleatório.

O modelo A descreve o caso aditivo, enquanto o modelo B e C envolvem respectivamente o caso multiplicativo da sazonalidade. Em geral assume-se que o índice sazonal S_t varia lentamente durante o tempo, de modo que $S_t \approx S_{t-s}$, onde s é o número de observações. A análise de séries temporais que apresentam variações sazonais dependem da necessidade de: 1. medir o efeito sazonal; 2. eliminar a sazonalidade. Para séries temporais que apresentam pouca tendência é geralmente adequado estimar o efeito sazonal em um período específico, por exemplo, o mês de setembro, onde encontra-se a média de cada observação de setembro, menos a média anual correspondente no caso aditivo, ou as observações de setembro divididas pela média anual no caso multiplicativo [7].

2.2.7 Autocorrelação

Antes de entrar neste tópico, é necessário introduzir o conceito de coeficiente de correlação. Coeficiente de correlação, denotado por r mede o grau da correlação entre duas variáveis. Normalmente r assume valores entre: $r = 1$ indica uma correlação perfeita positiva entre as duas variáveis.

$r = -1$ indica uma correlação negativa perfeita entre as duas variáveis, ou seja, quando uma aumenta a outra diminui.

$r = 0$ indica que as duas variáveis não são linearmente dependentes entre si.

O coeficiente de correlação é calculado de acordo com a fórmula 4:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (4)$$

onde x_1, x_2, \dots, x_n e y_1, y_2, \dots, y_n são os valores medidos de ambas as variáveis. Além disso também é preciso calcular as médias aritméticas das duas variáveis:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

O grau de correlação pode ser indicado:

0.7 a 1.0: demonstra uma forte correlação

0.3 a 0.7: demonstra uma correlação moderada.

0 a 0.3: demonstra uma fraca correlação.

Também observa-se a presença da correlação inversa, onde duas variáveis variam proporcionalmente em sentidos opostos:

-0.7 a -1.0: demonstra uma forte correlação inversa

-0.3 a -0.7: demonstra uma correlação inversa moderada

0 a -0.3: demonstra uma fraca correlação inversa.

Uma ideia similar pode ser aplicada a séries temporais para verificar se sucessivas observações estão correlacionadas. Dado N observações x_1, x_2, \dots, x_N , em uma série temporal discreta, forma-se $N - 1$ pares de observações, denominados: $(x_1, x_2), (x_2, x_3), \dots, (x_{N-1}, x_N)$. Considerando cada par como uma variável, o coeficiente de correlação entre x_t e x_{t+1} é definido pela equação 2.5:

$$r_1 = \frac{\sum_{t=1}^{N-1} (x_t - \bar{x})(x_{t+1} - \bar{x})}{\sum_{t=1}^N (x_t - \bar{x})^2} \quad (2.5)$$

onde \bar{x} é a média das observações obtida pela equação 2.6:

$$\bar{x} = \frac{1}{N} \sum_{t=1}^N x_t \quad (2.6)$$

A autocorrelação é um importante indicador das propriedades de uma série temporal, fornecendo um conjunto de coeficientes de autocorrelação que medem a correlação entre observações em diferentes distâncias. Estes coeficientes

geralmente oferecem ideias sobre o modelo probabilístico que gerou os dados. Autocorrelação é um tipo de cross-correlation do sinal consigo mesmo. Em geral é utilizado para encontrar padrões tais como a presença de alguma periodicidade que possa ter sido camuflada por um ruído. A autocorrelação de um processo aleatório descreve a correlação entre valores do processo em diferentes momentos de tempo, como uma função de dois tempos.

A equação 2.2. pode ser generalizada para calcular a correlação entre observações de k períodos anteriores de tempo, com a equação 2.2.7:

$$r_k = \frac{\sum_{i=1}^{n-k} (x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (2.7)$$

2.2.8 Correlograma

Um recurso útil para a interpretação de um conjunto de coeficientes de autocorrelação é a utilização de um gráfico chamado correlograma no qual r_k é projetado em relação ao atraso (lag) k .

No caso de uma série com flutuações sazonais, o correlograma exibirá uma oscilação na mesma frequência. Por exemplo, com observações mensais, r_6 será grande e negativo, enquanto r_{12} será grande e positivo (figura 2.4). Se x_t seguir um padrão senoidal, r_k também seguirá. Valores de r_k para longos atrasos, tendem a se aproximar de zero. Um exemplo pode ser visto na figura 2.4. Uma série temporal que dá origem a um correlograma desse tipo, é tipicamente uma na qual uma observação acima da média tende a ser seguida por uma ou várias observações acima da média, e também similarmente para observações abaixo da média.

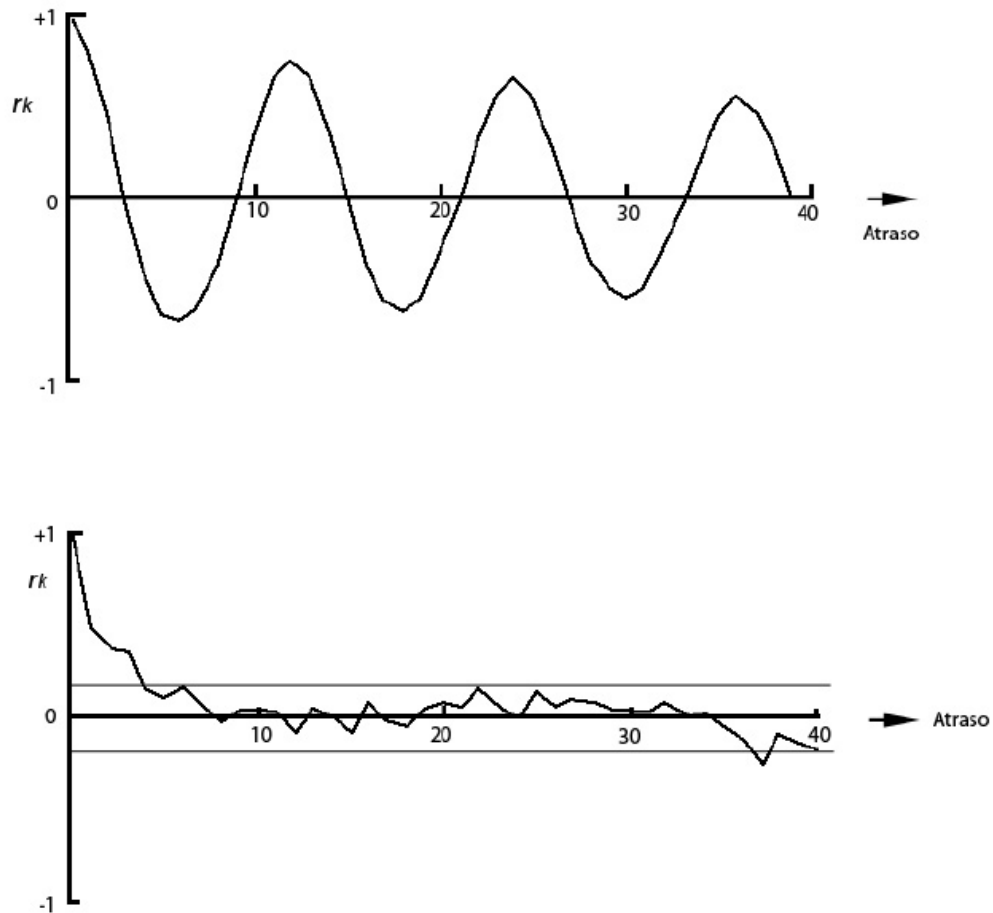


Figura 2.4 – Uma série temporal e seu respectivo correlograma.

2.2.9 Modelos Probabilísticos

Um processo estocástico pode ser descrito como um fenômeno estatístico que evolui no tempo de acordo com leis probabilísticas, sendo também conhecido como processo randômico, ou seja, um conjunto de variáveis aleatórias que estão ordenadas no tempo podendo ser contínuas ou discretas [33].

A análise de séries temporais está principalmente interessada em descobrir as propriedades do modelo probabilístico que gerou a série temporal observada.

Costuma-se descrever um processo estocástico através das funções média, variância e autocovariância:

$\mu(t)$ - média

$$\mu = \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (8)$$

$\sigma^2(t)$ - variância

$$\sigma^2 = VAR(X) = \frac{1}{(n-1)} \sum_{i=1}^n (x_i - \mu)^2 \quad (9)$$

$\gamma(t_1, t_2)$ - autocovariância: a função variância por si só não é suficiente para especificar os segundos momentos de uma sequência de variáveis aleatórias, i.e, precisa-se efetuar a medida da variação de duas variáveis, que neste caso é a medida da própria variável consigo mesma, em relação a um atraso k .

$$C_k = \sum_{i=1}^{n-k} \frac{(x_i - \mu)(x_{i+k} - \mu)}{(n-1)} \quad (10)$$

2.2.10 Modelo Autorregressivo

Frequentemente os valores de uma série temporal, em pontos específicos estão relacionados com valores passados e valores futuros. Uma autocorrelação de primeira ordem diz respeito a magnitude da relação entre valores consecutivos. Já uma autocorrelação de segunda ordem refere-se a magnitude da relação entre observações separadas por dois períodos. A p -ésima ordem de uma autocorrelação, refere-se ao tamanho da correlação entre valores que se encontram afastados por um intervalo de p períodos. A modelagem autorregressiva portanto tira vantagem da autocorrelação entre as observações para encontrar um melhor ajustamento histórico dos dados e desse modo realizar melhores prognósticos. Tal classe de método realiza a previsão de valores futuros, com base na autocorrelação presente entre duas observações de uma série temporal, distanciadas por um período p . Este modelo pode ser definido pela seguinte expressão:

$$X_t = \theta_0 + \theta_1 X_{t-1} + \theta_2 X_{t-2} + \dots + \theta_p X_{t-p} + \epsilon_t \quad (11)$$

onde,

X_t : é uma observação em um momento t (variável dependente);

X_{t-i} : a variável dependente em momentos $t - i$;

θ_p : são os coeficientes de autorregressão que correspondem aos parâmetros do modelo AR de ordem p ;

p : posição da autorregressão;

ϵ_t : representa o erro aleatório no momento t ;

Um modelo autorregressivo de primeira ordem considera somente a correlação entre valores consecutivos da série:

$$X_t = \theta_0 + \theta_1 X_{t-1} \quad (2.12)$$

Um modelo autorregressivo de segunda ordem considera a relação entre valores consecutivos e também a relação entre valores $t - 2$ (dois períodos).

$$X_t = \theta_0 + \theta_1 X_{t-1} + \theta_2 X_{t-2} \quad (2.13)$$

Para encontrar os coeficientes de ordem p de um modelo autorregressivo, pode-se utilizar a equação de Yule-Walker [34]:

$$B = (X^T X)^{-1} X^T Y \quad (2.14)$$

onde,

B : é o vetor de coeficientes

X : é a matriz contendo os valores dos dados observados, com um atraso t .

Y : é o vetor contendo os valores dos dados observados.

$$B = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_t \end{bmatrix} \quad Y = \begin{bmatrix} X_{t+1} \\ X_{t+2} \\ \vdots \\ X_n \end{bmatrix} \quad X = \begin{bmatrix} 1 & X_t & X_{t-1} & \dots & X_1 \\ 1 & X_{t+1} & X_t & \dots & X_2 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & X_{n-1} & X_{n-2} & \dots & X_{n-t} \end{bmatrix}$$

Existem outras técnicas para obtenção dos coeficientes do modelo autorregressivo, como podem ser vistos em [47], como por exemplo o método de Burg e o da covariância.

2.2.11 ARMA - Modelo Autoregressivo e Média Móvel

A metodologia Box e Jenkins é uma das técnicas mais tradicionais utilizadas na modelagem de análise de séries temporais devido a sua generalidade, podendo ser aplicada para uma diversidade de séries, estacionárias ou não [13]. Em geral, estes modelos são utilizados para avaliação de novos métodos sendo muito utilizados na comparação dos resultados com as redes neurais artificiais.

Um modelo ARMA consiste de duas partes: a parte autorregressiva (AR) e a parte de média móvel (MA), na qual especifica-se dois parâmetros (p, q) . O parâmetro p diz respeito a ordem da autorregressão e q é a ordem da média móvel.

Um modelo ARMA descreve uma série temporal de acordo com a equação 2.2.15 [42]:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \dots - \theta_q \epsilon_{t-q} \quad (2.15)$$

onde X_t é o valor previsto, X_{t-p} é um valor observado com um atraso p , ϵ_t é o termo que representa um erro no momento t , ϕ_i e θ_i são pesos associados as observações passadas e aos erros passados respectivamente.

A identificação de sistemas ARMA é um processo iterativo no qual busca-se identificar modelos com diferentes estruturas a partir dos dados observados, comparando-os em termos de desempenho. Se o desempenho do modelo é fraco, então aumenta-se gradualmente a complexidade de sua estrutura. Por último, seleciona-se aquele modelo que melhor descreve a dinâmica do sistema de modo mais simples. Um dos critérios adotado é o AIC - Akaike Information Criterion. O AIC ajuda a encontrar o número de defasagens (lags) do modelo ARMA, verificando a qualidade de ajuste dos dados pelo modelo e o número de parâmetros p e q . Essa qualidade é medida como uma função da variância dos resíduos dos erros: o ajuste melhora quando os resíduos tornam-se menores. Além disso o AIC penaliza aqueles modelos excessivamente complexos através do acompanhamento do número estimado de parâmetros do modelo. AIC pode ser calculado pela equação 2.2.16:

$$AIC = \log\left(V\left(1 + \frac{2n}{N}\right)\right) \quad (2.16)$$

onde V é a variância dos resíduos do modelo, N é o tamanho da série temporal e $n = (p + q)$ ou o número de parâmetros estimados pelo modelo. O AIC é calculado para vários candidatos sendo selecionado aquele que apresenta o menor AIC.

2.3 REDES NEURAIS

Nesta seção será apresentado brevemente a fisiologia do neurônio biológico cuja estrutura serviu de inspiração aos modelos de redes neurais artificiais (RNA). Além disso, será apresentado os conceitos fundamentais sobre redes neurais artificiais, o modelo de neurônio artificial, formas de aprendizagem e arquiteturas de RNAs.

2.3.1 Neurônio Biológico

Uma das principais causas do estudo da fisiologia cerebral dentro do campo da Inteligência Artificial (IA) está na premissa de que a capacidade de processamento de informações do cérebro é entendida como um produto, ou emerge, a partir da rede de neurônios e suas interconexões (sinapses) [30]. Desse modo o estudo do modelo neural biológico inspirou diversas pesquisas na IA, no intuito de entender e replicar em algum nível uma inteligência computacional, no sentido de simular em ambiente computacional características não determinísticas de resolução de problemas sem a definição ou especificação por parte dos seres humanos, possibilitando ao computador encontrar uma solução tipicamente através de algum processo de aprendizado ou inferência.

Um neurônio (figura 2.5) é uma unidade celular fundamental do sistema nervoso do cérebro, cuja principal função é processar, coletar e disseminar sinais elétricos, sendo extremamente estimulável e capaz de perceber as mínimas variações que ocorrem em torno de si, reagindo a estas variações com uma alteração elétrica que percorre sua membrana (impulso elétrico) [19]. Os neurônios

estabelecem entre si, conexões que permitem a transmissão de estímulos do ambiente. Um neurônio típico apresenta 3 partes distintas: corpo celular, dendritos e axônio. O corpo celular é a parte mais volumosa da célula nervosa, e é o local onde se encontra a maioria das estruturas citoplasmáticas. Ele é um elemento de processamento que recebe e combina sinais de outros neurônios vindos através de canais conhecidos como dendritos [26]. Dendritos são prolongamentos do neurônio especializados na recepção de estímulos. Quando a combinação dos estímulos que chegam no neurônio é forte o suficiente, o mesmo dispara, produzindo um sinal de saída ao longo do axônio. O axônio é um prolongamento da célula nervosa por onde se transmite os impulsos nervosos para os dendritos de outros neurônios. Cada sinal que chega a um neurônio pelo dendrito, passa através de uma sinapse. As sinapses funcionam como estruturas de interligação que medeiam as interações entre os neurônios e tipicamente operam da seguinte maneira [38]: um processo inicial libera uma substância química transmissora (neurotransmissor) que se propaga pela junção sináptica entre os neurônios, para então ocorrer o processo de conversão de um sinal elétrico (pré-sináptico) em um sinal químico e posteriormente em um sinal elétrico (pós-químico). Através do controle na frequência dos pulsos elétricos e com a inibição ou excitação da geração de novos pulsos em cada neurônio, as redes de sinapses são constituídas, fortalecidas ou enfraquecidas, sendo basicamente este processo que determina o comportamento e raciocínio no cérebro humano.

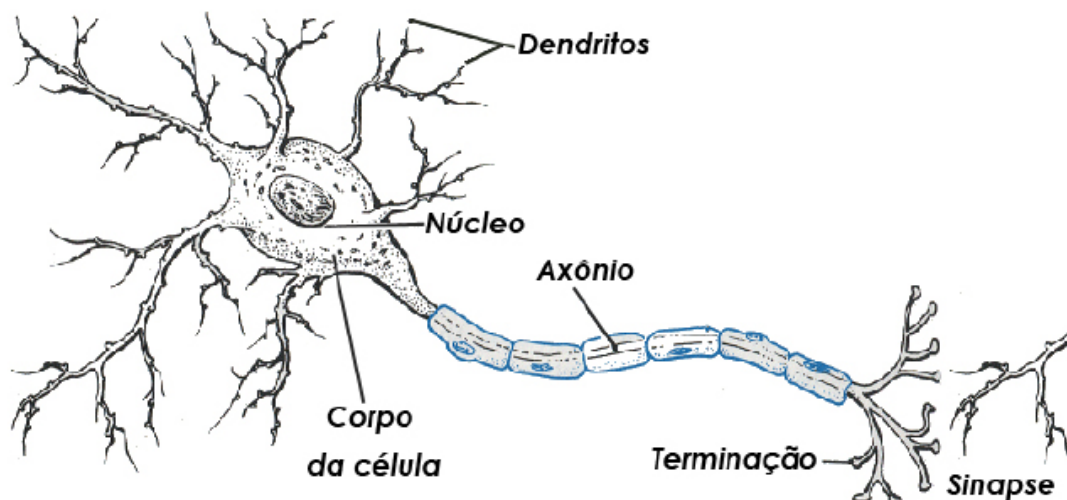


Figura 2.5 – Visão esquemática de um neurônio biológico [36]

2.3.2 Neurônio Artificial

O neurônio artificial é um modelo computacional inspirado no neurônio biológico, que busca simular parcialmente algumas características de sua contraparte biológica, dentro da capacidade tecnológica existente. A primeira parte de um neurônio artificial (figura 2.6) consiste na combinação das entradas, que chegam ao neurônio, vindas de outros neurônios através de uma função soma. Esta função gera um valor único a partir de um combinação linear ponderada pelos respectivos pesos sinápticos $(w_{0j}, w_{1j}, \dots, w_{ij}, \dots, w_{nj})$. A seguir este valor é submetido a uma função de ativação que mede o nível de atividade ou o estado do neurônio. Esta função visa restringir a amplitude de saída do neurônio. Esta amplitude é tipicamente normalizada num intervalo $[0,1]$ ou $[-1,1]$. Existem diversas funções de ativação que realizam esta medição, por exemplo: função limiar; função sigmóide; função threshold linear; função degrau e função gaussiana. Comparando um neurônio artificial com um neurônio biológico, pode-se por analogia dizer que: as conexões entre os elementos processadores de um neurônio artificial representam os axônios e dendritos; os pesos das conexões, representam as sinapses; a função soma, função de ativação e função saída representam a atividade do corpo do neurônio. A equação 2.2.17, mostra um exemplo de combinação linear no neurônio i , realizado pela função soma:

$$u_i = \sum_{j=0}^n W_{ji} X_j \quad (2.17)$$

onde X representa o vetor entrada e W representa a matriz de pesos.

A função de ativação é responsável pela transformação do vetor entrada, numa saída, sendo conhecida também como função de transferência. Ela funciona ao modo de um ajuste na importância da combinação das entradas de um neurônio, desempenhando dois papéis importantes na RNA: 1. deseja-se que a unidade seja ativa (aproxima-se de +1) quando as entradas forem corretas, ou mais próximas do resultado esperado; 2. seja inativa (aproxima-se de 0) quando as entradas erradas forem recebidas, ou aquelas que mais se distanciam do resultado desejado [30]. Um exemplo de função de ativação φ de um neurônio i pode ser vista na equação 2.2.18:

$$\varphi(u_i) = \varphi\left(\sum_{j=0}^n W_{ji} X_j\right) \quad (2.18)$$

Os sinais de entrada são representados por $x_0, x_1, x_2, \dots, x_n$. Cada uma dessas entradas é modificada por um peso associado (pesos sinápticos). Estes pesos podem ser tanto positivos como negativos, correspondendo tanto a aceleração ou inibição do fluxo dos sinais.

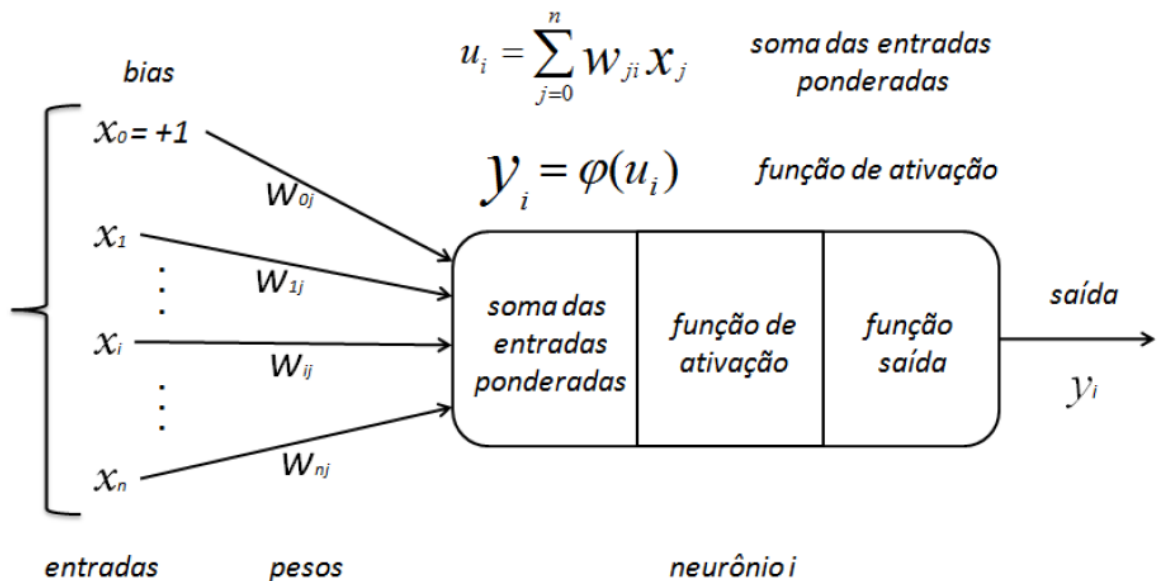


Figura 2.6 – Modelo de um neurônio artificial

O bias, representado por x_0 , é o parâmetro ($x_0 = +1$) da rede que define o limite real para uma unidade de processamento ser ativada. Isto ocorre quando a soma ponderada das entradas excede o valor definido no bias. Desse modo cada neurônio recebe um vetor de entradas e produz uma única saída.

Função de Ativação

Uma função de ativação ou função de transferência é uma função matemática, geralmente não-linear, que converte os valores de entrada de um neurônio para um valor geralmente baixo, restrito a um intervalo que normalmente varia entre $[-1, 1]$

A função de ativação é usada para converter a soma ponderada das entradas de um neurônio para um valor que represente a saída do neurônio. O quadro (2.1) apresenta 4 exemplos típicos de funções de transferência. Em relação a função sigmóide (figura 2.7), uma das principais vantagens é que ela é diferenciável em toda a sua extensão, o que a torna ideal para o algoritmo de atualização de pesos, sendo portanto uma função comumente utilizada pelo algoritmo backpropagation.

Função	Equação
Linear	$\varphi(u) = u$
Degrau	$\varphi(u) = \begin{cases} 1, & u \geq 0 \\ 0, & u < 0 \end{cases}$
Sigmóide	$\varphi(u) = \frac{1}{1+\exp(-u)}$
Tangente hiperbólica	$\varphi(u) = \frac{1-\exp^{-2u}}{1+\exp^{-2u}}$

Quadro 2.1- Exemplos de funções de transferência

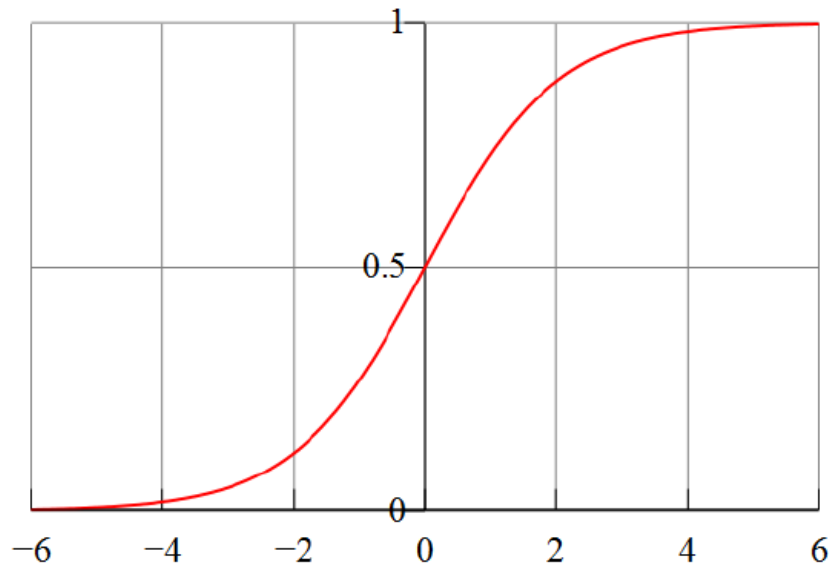


Figura 2.7 – Função sigmóide

2.3.3 Estrutura e Aplicações

Redes neurais têm sido utilizadas em diversas áreas, como na análise de séries temporais, processamento de sinais e reconhecimento de padrões, em virtude da sua capacidade de aprender a partir de um conjunto de dados de entrada com ou sem supervisão. Entretanto, conforme apontado em [8] e [38], a capacidade de funcionarem como aproximadores universais de funções é que tornam as RNAs uma técnica interessante para a classe de problemas de aproximação de funções.

RNAs são sistemas paralelos compostos por diversas unidades de processamento ou neurônios artificiais. Estas unidades de processamento realizam algum cálculo, sendo dispostas em camadas e interligadas por conexões, de modo análogo as sinapses, geralmente direcionais e com pesos numéricos associados. A complexidade da rede neural emerge a partir da interação de diversos neurônios artificiais.

De modo geral, pode-se definir 3 componentes básicos de uma RNA [24]:

- Neurônios: os nodos ou elementos de processamento da rede, tipicamente organizados em camadas, contendo: função de ativação, que representa o estado ou ativação do neurônio;
- Topologia ou arquitetura: um conjunto de sinapses ou canais de conexão entre os neurônios, com pesos associados. Uma conexão entre duas unidades i, j , serve para propagar a ativação da unidade j para a unidade i . Cada conexão tem um peso numérico associado w_{ji} , que determina a força e intensidade do sinal da conexão;
- Algoritmo de aprendizagem: algoritmo utilizado para efetuar o treinamento e ajuste dos pesos sinápticos.

Tipicamente uma RNA recebe um sinal de entrada, que consiste num vetor de números, x , que é apresentado à camada de entrada. Esse vetor sinal é transmitido pelas conexões para cada neurônio localizado na próxima camada, podendo ser amplificado ou atenuado de acordo com os pesos sinápticos (w_{ji}) associados a cada conexão.

Redes neurais são particularmente interessantes na resolução de problemas que não são expressos numa sequência de passos. Tipicamente RNAs podem ser bastante úteis no reconhecimento de padrões, classificação, previsão de séries temporais e data mining.

Segundo [14], redes neurais artificiais são particularmente interessantes na resolução de problemas da seguinte natureza:

Classificação: Envolve a identificação de um padrão, e a consequente associação do padrão em uma classe.

Categorização: Envolve a descoberta de categorias implícitas num conjunto de dados.

Aproximação: Envolve o aprendizado supervisionado, onde os padrões de entrada apresentados são mapeados para padrões de saída, através do ajuste dos pesos sinápticos.

Previsão: Envolve a questão da estimação de valores futuros, com base nos estados atuais e anteriores do sistema.

Otimização: Caracterizado pela minimização ou maximização de uma função custo, conhecida de antemão.

2.3.4 Arquitetura

A arquitetura de uma RNA compreende o modo como os neurônios estão organizados e distribuídos em camadas, e se estão: completamente interconectados; conectados parcialmente; ou não estão conectados. Além disso compreende também os padrões de conexão entre as camadas e dentro das mesmas.

Em [38] é apresentado 3 classes fundamentais de arquiteturas de redes:

- **Single-Layer Feed-forward Networks:** ou rede de perceptron, é a rede na qual os neurônios são organizados numa única camada. É a forma mais simples de redes em camadas, onde os padrões de entrada conectam-se diretamente a uma camada de saída de neurônios, sem camadas intermediárias, ou camadas ocultas.

- **Multilayer Feedforward Networks:** a segunda classe de redes neurais feedforward, diferencia-se pela presença de uma ou mais camadas escondidas, cujos nodos computacionais são denominados neurônios ocultos. Em geral, as entradas dos neurônios de uma camada são provenientes do sinal de saída dos neurônios da camada precedente, sem apresentar realimentação na camada de saída, ou seja, a saída produzida pela rede não é redirecionada como entrada na rede. A vantagem da utilização de uma camada oculta está no aumento da representação do espaço de hipóteses que a rede pode gerar, tornando a rede, assim, capaz de representar diferentes classes de domínios usando um hiperplano para separá-las [30]. Cada unidade da camada escondida, representa uma função limiar no espaço de entrada. A unidade de saída da rede funciona como uma combinação linear de cada uma dessas funções limiares.

A partir do uso de uma única camada oculta, é possível representar qualquer função contínua com relativa precisão; com duas camadas ocultas pode-se representar também funções descontínuas [30].

- **Recurrent Networks:** uma rede recorrente difere de uma rede feed-forward pelo fato de que os neurônio da camada de saída alimentam com seu sinal de saída, a entrada dos neurônio da camada de entrada, ou seja, nesse tipo de arquitetura, a saída produzida pela rede é usada para alimentar a entrada da própria rede, formando um fluxo de realimentação contínua.

2.3.5 Aprendizagem

A aprendizagem é a etapa na qual a rede ajusta os pesos sinápticos de modo a encontrar uma relação entre cada um dos pares entrada-saída do conjunto exemplo. Essa etapa é realizada após a definição da arquitetura da RNA.

A aprendizagem ocorre de modo sistemático, onde os parâmetros livres da rede vão sendo ajustados progressivamente a cada iteração pelo conjunto exemplo. Estes parâmetros livres são: os pesos sinápticos e os níveis de bias.

O conhecimento é representado e armazenado no conjunto de pesos sinápticos calibrados a partir de um conjunto de exemplos, previamente

apresentados para a rede. Uma memória associativa é o princípio que rege o processo de aprendizagem de uma rede [38].

Tipicamente os algoritmos de aprendizagem de uma RNA podem ser classificados em duas categorias:

- **supervisionado:** a aprendizagem supervisionada, ou aprendizagem com professor, consiste na ação de indicar a saída desejada para um padrão de entrada, efetuando, para isso, ajustes nos parâmetros da rede para que a mesma encontre uma relação entre os pares entrada-saída. Parte-se do princípio que o “professor” tem conhecimento sobre o ambiente, sendo este conhecimento representado pelo conjunto de exemplos entrada-saída [38].

- **não-supervisionado:** inexistência de uma influência externa indicando qual a resposta desejada para um conjunto de entradas. Em geral utilizado para agrupar os dados de entrada em categorias através de neurônios classificadores. Consiste desse modo de um processo de descoberta dos grupos de classificação no qual as entradas serão associadas.

2.3.6 Perceptron

Em 1958 Rosenblatt [38] propôs um algoritmo de aprendizagem para ajustar os pesos de uma rede neural que consistia de apenas um neurônio. Esta rede ficou conhecida como o perceptron de Rosenblatt. O perceptron constituído de apenas um único neurônio, consegue classificar padrões de apenas duas classes distintas, ou seja, apenas padrões que podem ser separados por um hiperplano. A função de ativação limita tipicamente a saída dos neurônios a -1, 0 ou 1. A regra de aprendizado de uma rede perceptron num momento n consiste basicamente em encontrar o valor do incremento $\Delta w(n)$ para ser aplicado na atualização dos pesos, ou na formulação da equação de ajuste: $w(n+1) = w(n) + \Delta w(n)$. Lembrando que o algoritmo de treinamento perceptron é aplicável apenas a classes de problemas linearmente separáveis, apresenta-se abaixo, de maneira geral, a descrição do algoritmo de treinamento do perceptron simples:

1. Definir o valor inicial de α ;
2. Iniciar o vetor de pesos w , com valores gerados aleatoriamente.

3. Aplicar a equação de ajustes de pesos para todos os elementos do conjunto exemplo: $w(n+1) = w(n) + \alpha ex(n)$, onde α é a taxa de aprendizagem, e representa o erro ($e = y_d - y$, a diferença da entrada desejada com a entrada calculada), x é o vetor entrada e w o vetor de pesos.

4. Repetir o processo até $e = 0$, para todos os pares de treinamento.

Adaline

O modelo Adaline também é baseado em unidades de processamento com função soma das entradas ponderadas. Utiliza o algoritmo de treinamento conhecido como regra delta (19), proposto por Widrow e Hoff. Este algoritmo compara a saída gerada pela rede em relação ao erro obtido pela diferença entre a saída desejada e a saída gerada. Assim o ajuste dos pesos é feito de acordo com a magnitude do erro. O algoritmo se baseia principalmente na magnitude do sinal do gradiente do erro, que funciona como direção para o ajuste do peso ou Δw .

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n) \quad (19)$$

sendo,

$w_{ji}(n+1)$ - o novo valor do peso, após sofrer correção
 $w_{ji}(n)$ - o valor anterior do peso
 $\Delta w_{ji}(n)$ - a mudança que será incrementada no peso sináptico de acordo com o erro.

A principal diferença entre o perceptron e o modelo Adalaine é que o perceptron funciona como um separador linear, podendo ser utilizado para classificação, enquanto que o Adaline funciona como um aproximador linear de funções [14].

2.3.7 Multi-Layer Perceptron

No caso de redes tipo MLP os neurônios são organizados numa sequência de camadas. Cada camada contém portanto um grupo de neurônios. Na MLP existe a camada de entrada, que recebe o vetor de dados de entrada; a camada de saída que contém neurônios cujas saídas representam a resposta da

rede de acordo com a entrada recebida; e uma ou mais camadas ocultas que tem por finalidade ampliar a flexibilidade da rede em representar problemas não lineares [55].

O mecanismo básico consiste em treinar esta rede com dados reais, ou seja, dados empíricos, com amostras ou exemplos de padrões de entrada-saída, para construir uma relação não-linear entre os exemplos e a partir disso, criar um modelo que seja generalista e permita realizar previsões.

Uma rede MLP (figura 2.8) tem tipicamente 3 características distintas:

1. A função de ativação utilizada no processamento interno do neurônio é não-linear e diferenciável em toda sua extensão. Uma função normalmente utilizada é a função sigmóide (20).

$$y_j = \frac{1}{1 + \exp(-v_j)} \quad (20)$$

2. A existência de uma ou mais camadas ocultas.
3. Os neurônios entre camadas vizinhas são densamente interligados.

Cada camada possui um conjunto de neurônios. Na camada de entrada estão localizados os neurônios de entrada responsáveis por receberem os padrões que serão apresentados a RNA. Logo em seguida tem-se os neurônios ocultos localizados na camada intermediária, ajudando a estabelecer a relação não-linear entre entrada e saída [37]. Por último estão os neurônios de saída (camada de saída) que representam a saída gerada pela rede, cujo valor é comparado com a saída desejada durante a fase de aprendizagem. O número de neurônios nesta camada também estão relacionados ao horizonte de previsão [37].

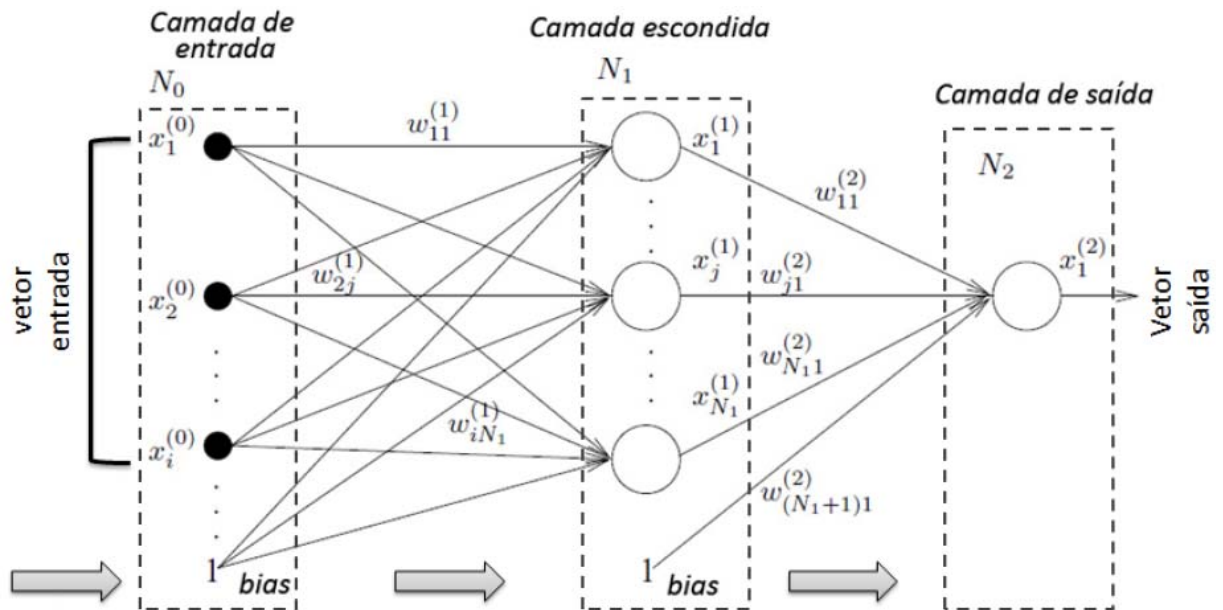


Figura 2.8 – Uma rede MLP com 1 camada oculta

Conforme a figura (2.8) observa-se que o fluxo dos dados de entrada seguem uma única direção, ou seja, iniciam a partir da camada de entrada, passam pelas camadas intermediárias e terminam na camada de saída. Nesse cenário a saída da rede não é utilizada para realimentar a entrada.

A MLP é particularmente interessante para separar diferentes classes que não são possíveis de serem separadas por um hiperplano, ou seja, são úteis para a classificação de padrões não linearmente separáveis. A introdução de uma camada oculta na arquitetura da rede, possibilita que a mesma adquira capacidade para resolver problemas não linearmente separáveis.

Em seu caso mais simples, a MLP consiste de 3 camadas (figura 2.8): 1. a camada de entrada, por onde a rede recebe a informação do ambiente através de unidades sensoriais; 2. a camada oculta de nós computacionais; 3. a camada de saída de nós computacionais [38]. Os sinais de entrada da rede, são propagados em um único sentido, na direção camada de entrada, camada oculta e por último, camada de saída.

O conjunto dos pesos sinápticos incidentes em cada neurônio, são estruturados e representados através de uma matriz de pesos, que funciona como a memória da rede neural, permitindo a rede “lembrar” de determinados padrões

quando uma entrada é apresentada. No caso da RNA na figura (2.8) uma típica matriz de pesos pode ser descrita da seguinte forma:

$$\mathbf{W} = \begin{vmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2n} \\ w_{31} & w_{32} & w_{33} & \dots & w_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ w_{n1} & w_{n2} & w_{n3} & \dots & w_{nn} \end{vmatrix}$$

onde o peso w_{ij} representa o peso j , incidente no neurônio i .

Aprendizagem MLP

Durante a realização de um ciclo de aprendizagem na RNA tem-se tipicamente a seguinte sequência de eventos:

- Padrões provenientes de um ambiente são submetidos como entrada na rede, estimulando-a.
- Essa estimulação gera eventos internos na RNA, que iniciam mudanças internas, em seus parâmetros livres.
- As mudanças efetivadas, em princípio, tornam a rede melhor adaptada para o ambiente, permitindo que a mesma responda de um modo novo aos estímulos externos.

Tipicamente a aprendizagem de uma rede MLP é supervisionada, utilizando em geral, a técnica da minimização de uma função erro, aplicando as modificações nos pesos sinápticos de acordo com o cálculo dessa função. Inicia-se com a criação de uma matriz de pesos com valores aleatórios. Em seguida, para cada elemento do conjunto exemplo, o algoritmo de aprendizagem efetua os ajustes na matriz de pesos, de modo a atualizar essa matriz com a contribuição de cada elemento do conjunto exemplo (figura 2.9). O erro de saída indica o quão distante ou diferente as saídas da rede neural estão da saída desejada ou ideal. Outro critério também utilizado é o número de épocas (iterações) realizadas sobre o conjunto exemplo. Em alguns casos, a convergência não ocorre, e portanto, a minimização do erro não é possível de ser realizada. Nesse caso, o número de épocas funciona como critério alternativo de parada.

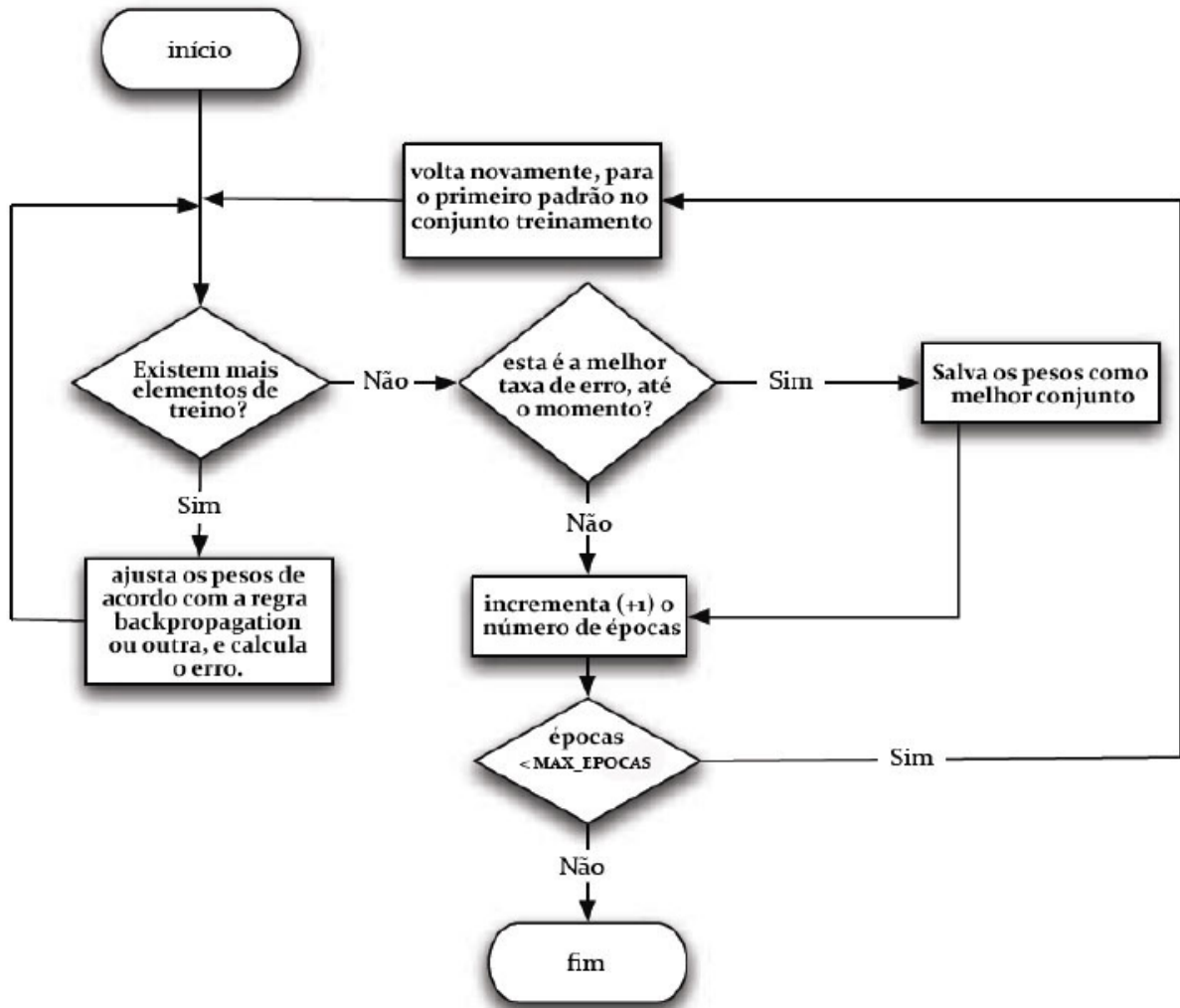


Figura 2.9 – Aprendizagem supervisionada, adaptado de [41]

A aprendizagem em uma RNA envolve duas fases importantes:

- Fase de treinamento: o objetivo da fase de treinamento, ou calibração, é alcançar uma solução ótima, com base em medidas de desempenho tais como o RMSE (equação 4.2) [32]. Esta etapa exige em geral um acompanhamento, ou aprendizagem supervisionada, para verificar se a RNA está generalizando ou simplesmente memorizando os dados.
- Fase de teste: consiste em usar um conjunto diferente daquele que foi usado para o treinamento, mas ainda com valores de saída conhecidos e analisar a taxa de acerto deste conjunto.

Durante a etapa de aprendizagem um dos objetivos é efetuar a calibração dos pesos de modo que a RNA consiga desenvolver a capacidade de generalização. De acordo com [38], generalização refere-se a capacidade da RNA produzir dados de saída razoáveis para entrada de dados não especificadas durante

a fase de treinamento (aprendizagem).

A generalização constitui-se na etapa que ocorre durante a aprendizagem, onde a rede deve ser capaz de gerar respostas adequadas para entradas que não estiveram presentes durante a fase de treinamento. Esse processo deve ser supervisionado para verificar se a rede não está apenas memorizando os dados.

De acordo com o andamento do aprendizado, a rede vai aprendendo o modelo ou a função subjacente, que melhor representa o domínio dos dados de treinamento. Uma importante aspecto nessa etapa é a rede manter a sua capacidade de resposta para as mesmas entradas e além disso ser flexível o suficiente para aprender novas entradas de maneira que as entradas anteriores não sejam sobre-escritas pelos novos padrões de dados apresentados.

A validação de uma rede neural é uma etapa final importante para verificar se a rede conseguiu atingir um grau aceitável de generalização, e se precisa de mais treinamento. Para a validação correta da rede, os dados utilizados devem ser diferentes daqueles usados na fase de treinamento.

A validação também pode evidenciar problemas estruturais na rede, tais como: número de neurônios inadequados; conjunto treinamento insuficiente; problema de memorização dos padrões; taxa de aprendizagem inadequada; número de camadas inapropriadas; escolha do conjunto de pesos iniciais mal estimada; convergência para um mínimo local.

Existem três parâmetros livres que exercem influência determinante na convergência da rede neural, em específico no algoritmo backpropagation padrão (BP):

- **Taxa de aprendizagem:** valor que controla o tamanho do ajuste realizado na correção dos pesos durante o processo de treinamento. Essa taxa oscila no intervalo $[0:1]$, sendo que, quanto maior a taxa, maior o percentual de ajuste dos pesos, levando a um aprendizado mais rápido, entretanto podendo também levar a uma instabilidade na convergência da rede.

- **Momentum:** parâmetro que auxilia o backpropagation a não cair num mínimo local raso, adicionando uma fração m ao valor de atualização do peso anterior. O momentum também é um valor definido no intervalo $[0:1]$, e permite que seja assimilado as atualizações passadas realizadas nos pesos, nas próximas atualizações, melhorando a convergência e diminuindo as oscilações. Um

momentum próximo do valor 1, significa que os pesos estão sendo atualizados levando-se em consideração as atualizações passadas. Essa estratégia aumenta a velocidade de convergência em regiões de descida da superfície de erro, devido justamente ao incremento de parcelas anteriores do erro na fórmula de ajustes de pesos [14].

- **Valor inicial dos pesos:** a atribuição inicial dos valores das conexões sinápticas pode levar a uma estagnação da rede num mínimo local.

2.3.8 Algoritmo Backpropagation

Existem diversos algoritmos de aprendizagem para treinar as redes neurais, conforme discutido em [12]. Em geral, eles trabalham de alguma maneira na modificação da matriz de pesos, e diferem entre si, no modo como é realizado esse ajuste nos pesos.

Willians, Rumelhart [21] desenvolveram em 1986 a classe de algoritmo (backpropagation ou retropropagação) para efetuar o ajuste de pesos em redes com camadas ocultas, tornando-se o algoritmo mais popular para o treinamento de redes MLP. Esse tipo de aprendizagem ocorre por correção do erro, onde tipicamente busca-se a minimização do mesmo, no resultado global produzido pela RNA.

A aprendizagem por retropropagação do erro inicia-se pela propagação de um padrão de entrada, através das camadas da rede, até atingir a camada de saída. Em seguida ocorre o processo inverso, ou seja, a retropropagação, onde os valores gerados pela camada de saída são passados para as camadas anteriores, sendo que a cada passo para trás, os pesos sinápticos vão sendo ajustados de acordo com o cálculo de erro (a diferença entre a saída desejada e a saída obtida, sendo que no caso de camadas ocultas, a saída desejada precisa ser calculada).

O backpropagation é um algoritmo de aprendizagem interessante quando se tem uma ou mais camadas ocultas na rede. A questão é que enquanto o erro na camada de saída é bem definido, através da diferença entre valor obtido e valor observado, nos neurônios ocultos ele não é, pois os dados de treinamento não

informam que valores os nodos ocultos devem ter como saída. Através da propagação de retorno, ou seja, a propagação do erro da camada de saída para a camada oculta, é possível fazer o ajuste necessário na camada oculta. A ideia é que cada neurônio da camada oculta tem uma parcela de contribuição no erro global da rede. Assim, os pesos associados a cada conexão devem ser atualizados de acordo com a intensidade da conexão entre o nó oculto e o nó de saída. Esse procedimento é realizado quando obtido uma solução que não satisfaça os critérios de desempenho da rede. Nesse caso, o algoritmo de backpropagation, faz o caminho contrário, ajustando os pesos sinápticos de cada aresta. Esse processo é repetido até que a saída satisfaça algum critério de desempenho, como por exemplo a minimização do erro de saída da rede.

Segundo [14] a formulação das equações de ajuste dos pesos do algoritmo backpropagation de um neurônio j que possui uma saída linear u_j , correspondente à soma ponderada de suas entradas, e uma saída, normalmente não-linear, y_j obtida após a aplicação da função de ativação sobre u_j , ou seja, $y_j = f(u_j)$. Para diferenciar as respostas dos neurônios das camadas de saída e escondida, estes últimos terão suas saídas referenciadas como $h(u_i)$ para um neurônio i qualquer.

O erro de saída de um neurônio j na iteração n é $e_j(n) = y_d^j(n) - y_j(n)$, onde a soma dos erros quadráticos de todos os neurônios de saída é definido na equação 2.2.21

$$\epsilon(n) = \frac{1}{2} \sum_j e_j^2(n) \quad (2.21)$$

Sendo a saída do neurônio j da camada de saída definida por $u_j = \sum_i h_i(n)w_{ji}(n)$, onde i é o índice referente à camada escondida, reescreve-se o erro do neurônio j como $e_j(n) = y_d^j(n) - f(u_j(n))$. Desse modo a equação 2.2.21 pode ser reescrita na equação 2.2.22.

$$\epsilon(n) = \frac{1}{2} \sum_j (y_d^j(n) - f(u_j(n)))^2 \quad (2.22)$$

Camada de saída

Para a camada de saída, o vetor de pesos deve ser ajustado em direção contrária ao gradiente do erro. Assim, as derivadas parciais de ϵ em relação

a cada um dos pesos da camada de saída devem ser obtidas. Para o neurônio j , a derivada parcial de ϵ em relação ao peso w_{ji} que o conecta ao neurônio i da camada escondida pode ser obtida pela equação 2.2.23

$$\frac{\partial \epsilon(n)}{\partial w_{ji}} = \frac{1}{2} \frac{\partial}{\partial w_{ji}} (y_d^j(n) - f(u_j(n)))^2 \quad (2.23)$$

Aplicando-se a regra da cadeia obtém-se a equação 2.2.24:

$$\frac{\partial \epsilon(n)}{\partial w_{ji}} = \frac{1}{2} 2(y_d^j(n) - f(u_j(n))) \frac{\partial}{\partial w_{ji}} (y_d^j(n) - f(u_j(n))) \quad (2.24)$$

Sendo que $e_j = (y_d^j(n) - f(u_j(n)))$, obtem-se a equação 2.2.25

$$\frac{\partial \epsilon(n)}{\partial w_{ji}} = e_j(n) \frac{\partial}{\partial w_{ji}} (y_d^j(n) - f(u_j(n))) \quad (2.25)$$

E, novamente, utilizando a regra da cadeia para chegar na derivada $\frac{\partial}{\partial w_{ji}} (y_d^j(n) - f(u_j(n)))$, obtem-se às equações 2.2.26 e 2.2.27

$$\frac{\partial \epsilon(n)}{\partial w_{ji}} = e_j(n) \frac{\partial}{\partial u_j} (y_d^j(n) - f(u_j(n))) \frac{\partial}{\partial w_{ji}} u_j(n) \quad (2.26)$$

$$\frac{\partial \epsilon(n)}{\partial w_{ji}} = e_j(n) (-1) \frac{\partial}{\partial u_j} f(u_j(n)) \frac{\partial}{\partial w_{ji}} \sum_k h_k(n) w_{jk}(n) \quad (2.27)$$

Sendo que no somatório $\sum_k h_k(n) w_{jk}(n)$ apenas o termo em que $k = i$ não terá derivada nula, tem-se que $\frac{\partial}{\partial w_{ji}} \sum_k h_k(n) w_{jk}(n) = h_i(n)$. A derivada $\frac{\partial}{\partial u_j} f(u_j(n))$ pode ser representada por $f'(u_j(n))$, que corresponde à derivada da função de ativação do neurônio j em relação ao valor de u_j . Obtem-se assim, a equação 2.2.28 para o ajuste dos pesos de um neurônio j qualquer da camada de saída.

$$\frac{\partial \epsilon}{\partial w_{ji}}(n) = -e_j(n) f'(u_j(n)) h_i(n) \quad (2.28)$$

Camada Escondida

Supondo que k refere-se a uma entrada da rede de duas camadas, a derivada parcial do erro de saída ϵ em relação ao peso w_{ik} pode ser representada pela equação 2.2.29, na qual o somatório ocorre sobre todos os neurônios j de saída.

$$\frac{\partial \epsilon}{\partial w_{ik}}(n) = \frac{\partial}{\partial w_{ik}} \frac{1}{2} \sum_j (y_d^j(n) - f(\sum_i h_i(n) w_{ji}(n)))^2 \quad (2.29)$$

O somatório pode ser reescrito de acordo com a equação 2.2.30, como a soma dos erros de cada um dos neurônios,

$$\frac{\partial \epsilon}{\partial w_{ik}}(n) = \frac{\partial}{\partial w_{ik}} \frac{1}{2} (e_1^2(n) + e_2^2(n) + e_3^2(n) + \dots + e_m^2(n)) \quad (2.30)$$

sendo m o número de neurônios da camada de saída. A seguir, a derivada do somatório pode ser representada pela equação 2.2.31

$$\frac{\partial \epsilon}{\partial w_{ik}}(n) = \frac{1}{2} \left(\frac{\partial}{\partial w_{ik}} e_1^2(n) + \frac{\partial}{\partial w_{ik}} e_2^2(n) + \frac{\partial}{\partial w_{ik}} e_3^2(n) + \dots + \frac{\partial}{\partial w_{ik}} e_m^2(n) \right) \quad (2.31)$$

Analisando cada um dos termos do somatório e obtendo-se a derivada para cada termo j , chega-se a equação 2.2.32

$$\frac{\partial e_j^2}{\partial w_{ik}}(n) = \frac{\partial}{\partial w_{ik}} (y_d^j(n) - f(u_j(n)))^2 \quad (2.32)$$

Obtendo-se a derivada parcial da equação 2.2.32, chega-se as equações 2.2.33 e 2.2.34:

$$\frac{\partial e_j^2}{\partial w_{ik}}(n) = (2)(-1)e_j(n)f'(u_j(n))\frac{\partial}{\partial w_{ik}}u_j(n) \quad (2.33)$$

$$\frac{\partial e_j^2}{\partial w_{ik}}(n) = (-2)e_j(n)f'(u_j(n))\frac{\partial}{\partial w_{ik}}u_j(n) \quad (2.34)$$

Entretanto, $u_j(n)$ corresponde ao somatório das contribuições dos neurônios i conectados a j , a derivada $\frac{\partial}{\partial w_{ik}}u_j(n)$ pode ser obtida pela equação 2.2.35.

$$\frac{\partial}{\partial w_{ik}}u_j(n) = \frac{\partial}{\partial w_{ik}} \left(\sum_i h_i(n)w_{ji}(n) \right) \quad (2.35)$$

Como somente o neurônio i da camada escondida tem o peso w_{ik} como entrada, a derivada do somatório pode ser representada nas equações 2.2.36 e 2.2.37

$$\frac{\partial}{\partial w_{ik}}u_j(n) = \frac{\partial}{\partial w_{ik}} (w_{ik}(n)h_i(n)) \quad (2.36)$$

$$\frac{\partial}{\partial w_{ik}}u_j(n) = w_{ji}(n)\frac{\partial}{\partial ik}h_i(n) \quad (2.37)$$

Aplicando-se a regra da cadeia,obtem-se a equação 2.2.38

$$\frac{\partial}{\partial w_{ik}}u_j(n) = w_{ji}(n)h'_i(u_i(n))\frac{\partial}{\partial w_{ik}}u_i(n) \quad (2.38)$$

Sendo $u_i(n)$ o termo que corresponde a soma ponderada das

entradas do neurônio j , a derivada $\frac{\partial}{\partial w_{ik}} u_i(n)$ fica apenas $x_k(n)$, pois todos os termos do somatório são constantes, exceto $w_{ik}x_k(n)$, obtendo-se assim a equação 2.2.39.

$$\frac{\partial}{\partial w_{ik}} u_j(n) = w_{ji}(n) h'_i(u_i(n)) x_k(n) \quad (2.39)$$

Na equação 2.2.34, substituindo-se pela equação 2.2.39 obtém-se a equação 2.2.40 a seguir:

$$\frac{\partial e_j^2}{\partial w_{ik}}(n) = (-2)e_j(n) f'(u_j(n)) w_{ji}(n) h'_i(u_i(n)) x_k(n) \quad (2.40)$$

Substituindo-se agora a Equação 2.2.40 na Equação 2.2.30, obtemos a Equação 2.2.41

$$\begin{aligned} \frac{\partial \epsilon}{\partial w_{ik}(n)} = & \frac{1}{2}((-2)e_1(n) f'(u_1(n)) w_{1i}(n) h'_i(u_i(n)) x_k(n) + \\ & (-2)e_2(n) f'(u_2(n)) w_{2i}(n) h'_i(u_i(n)) x_k(n) + \\ & (-2)e_3(n) f'(u_3(n)) w_{3i}(n) h'_i(u_i(n)) x_k(n) + \dots + \\ & (-2)e_m(n) f'(u_m(n)) w_{mi}(n) h'_i(u_i(n))) \end{aligned} \quad (2.41)$$

Colocando-se em evidência o termo $h'_i(u_i(n)) x_k(n)$, obtém-se a equação 2.2.42

$$\begin{aligned} \frac{\partial \epsilon}{\partial w_{ik}(n)} = & \frac{1}{2}(-2) h'_i(u_i(n)) x_k(n) (e_1(n) f'(u_1(n)) w_{1i}(n) + \\ & e_2(n) f'(u_2(n)) w_{2i}(n) + \\ & e_3(n) f'(u_3(n)) w_{3i}(n) x_i(n) + \dots + \\ & e_m(n) f'(u_m(n)) w_{mi}(n)) \end{aligned} \quad (2.42)$$

Escrevendo os termos dentro do parênteses na forma de somatório, obtém-se equação 2.2.43.

$$\frac{\partial \epsilon}{\partial w_{ik}}(n) = -h'_i(u_i(n)) x_k(n) \sum_j e_j(n) f'(u_j(n)) w_{ji}(n) \quad (2.43)$$

Como o ajuste dos pesos deve ser feito na direção contrária ao gradiente, a equação 2.2.44 apresenta o ajuste a ser aplicado no peso arbitrário w_{ik} , que conecta a entrada k ao neurônio i da camada escondida,

$$\Delta w_{ik}(n) = \alpha h'_i(u_i(n)) \sum_j e_j(n) f'(u_j(n)) w_{ji}(n) x_k(n) \quad (2.44)$$

onde α é uma constante de proporcionalidade correspondente à taxa

de aprendizado.

- $\alpha h'_i(u_i(n))$ - corresponde à derivada da função de ativação do neurônio i da camada escondida.
- $u_i(n)$ - corresponde à soma ponderada das entradas do neurônio i
- $\sum_j e_j(n) f'(u_j(n)) w_{ji}(n)$ - corresponde a uma medida de erro do neurônio i da camada escondida. Como o somatório é feito em j , correspondendo aos neurônios da camada de saída, tem-se aqui a soma ponderada de todos os erros dos neurônios de saída pelos pesos que os conectam ao neurônio i da camada escondida. Através dessa soma ponderada dos erros da camada de saída, os erros calculados com base no conjunto treinamento voltam para trás para permitir o ajuste dos neurônios da camada escondida.

2.4 ALGORITMOS GENÉTICOS

Algoritmos genéticos são baseados nos princípios da teoria evolucionista de Darwin. Nesta teoria um componente chave da evolução é a seleção natural, ou seja, indivíduos mal adaptados ao seu ambiente tendem a se extinguir, enquanto que aqueles mais bem adaptados sobrevivem. Estes organismos sobreviventes geram proles que possuem em seus materiais genéticos, as características que auxiliaram seus ancestrais a sobreviverem. Como resultado os descendentes tendem a adaptarem-se melhor ao ambiente, possuindo portanto, melhores chances de sobrevivência, acasalamento e geração de descendentes.

Algoritmo genético (AG) é uma sub-área da Computação evolutiva, sendo uma interessante ferramenta de busca e otimização para diversas classes de problemas [52]. Num cenário onde existem um grande número de possíveis soluções para um problema, torna-se impraticável encontrar a solução ótima através de algoritmos iterativos, sendo os AGs, nesse caso, uma alternativa para buscas amplas nesse espaço de estados [41].

Os AGs operam sobre uma população de candidatos ou indivíduos. Cada indivíduo representa uma possível solução no espaço de busca. Segundo [52], os AGs diferem dos métodos tradicionais de busca e otimização, em 4 aspectos

principais:

1. utilizam uma codificação do conjunto de parâmetros.
2. utilizam uma população de elementos ao invés de um único elemento.
3. utilizam informações de custo, ou seja o quão boa é uma solução.
4. utilizam regras com base em probabilidades, não sendo portanto determinístico.

Em geral o funcionamento do AG inicia-se com a criação de uma população constituída por um conjunto de indivíduos criados aleatoriamente, representando possíveis soluções do problema em questão. Essa população é avaliada por uma função custo, onde cada indivíduo recebe uma nota ou índice que reflete a sua habilidade de adaptação ao ambiente. Um percentual de indivíduos mais aptos é mantida, enquanto os outros (menos aptos) são descartados. Aqueles que são mantidos pela seleção, podem passar por modificações através de mutação e recombinação (cruzamento) gerando descendentes para a próxima geração, que será a nova população de indivíduos avaliados pelo algoritmo genético [52]. Aqueles menos adaptados tenderão a extinção conforme as novas gerações de populações vão sendo formadas. Os cromossomos gerados por reprodução são adicionados a antiga população, substituindo aqueles com pior função de ajuste. A partir desse momento, tem-se uma nova população.

2.4.1 Cromossomo

Um cromossomo representa uma possível solução do problema modelado pelo AG. Cada cromossomo possui diversos genes, que codificam uma característica da solução. Existem diversas modalidades de codificação, entretanto pode-se citar 3 tipos comumente encontrados:

1. **Binária:** onde 0 e 1 podem ser utilizados para indicar a presença ou ausência de uma propriedade. No exemplo da figura 2.10 tem-se um cromossomo com codificação binária, onde cada gene assume apenas os valores 0 ou 1. Cada gene pode representar parte da solução, ou a existência ou não de uma

característica.

Cromossomo A	0	1	1	0	1	0
---------------------	---	---	---	---	---	---

Figura 2.10 – Exemplo de cromossomo com codificação binária

2. **Valor:** os genes são números reais, isto é, podem assumir qualquer valor real, tanto positivo quanto negativo. A figura 2.11 mostra um exemplo desse tipo de codificação. Nesse caso, o operador mutação deve ser definido de acordo com o problema em si modelado.

Cromossomo A	2	0,4	1	4,1	3	1,2
---------------------	---	-----	---	-----	---	-----

Figura 2.11 – Exemplo de cromossomo com codificação por valor

3. **Permutação:** os genes representam uma possível sequência/caminho para percorrer. Conforme a figura 2.12 apresenta, cada gene pode representar um estado, no qual, o cromossomo pode representar uma rota ou caminho a ser percorrido. Nesse tipo de codificação deve-se ter atenção quanto ao tipo de mutação e quanto ao cruzamento, pois quando se trabalha com caminhos, as rotas não podem repetir os mesmos genes.

Cromossomo A	A	B	F	E	C	D
---------------------	---	---	---	---	---	---

Figura 2.12 – Exemplo de cromossomo com codificação por permutação

Portanto, nos AGs um indivíduo bem adaptado é na verdade um cromossomo que apresenta um bom ajuste, calculado por uma função objetivo, que tipicamente se deseja maximizar ou minimizar.

2.4.2 Seleção

Depois de decidido o tipo de codificação dos cromossomos é preciso definir como será feita a seleção dos cromossomos de uma população para a geração dos descendentes ou da próxima geração. A seleção consiste na etapa na qual o AG escolhe indivíduos de uma população para reprodução. A seleção busca priorizar os indivíduos mais aptos, entretanto deve-se ter um balanceamento variando a mutação e a recombinação, nesse caso como estratégia de heurística para exploração do espaço de estados.

2.4.3 Recombinação

A recombinação ocorre a partir da definição de um ponto de seccionamento. A prole será composta de parte dos genes do “pai” e da “mãe”. Essa parte é definida pelo ponto de corte. Por exemplo, na figura 2.13, é definido um ponto de corte na terceira casa, sendo portanto, os descendentes formados pela cópia dos genes do cromossomo A até a terceira casa, e o restante do cromossomo B. O segundo descendente também é uma cópia, entretanto na ordem inversa do primeiro descendente.

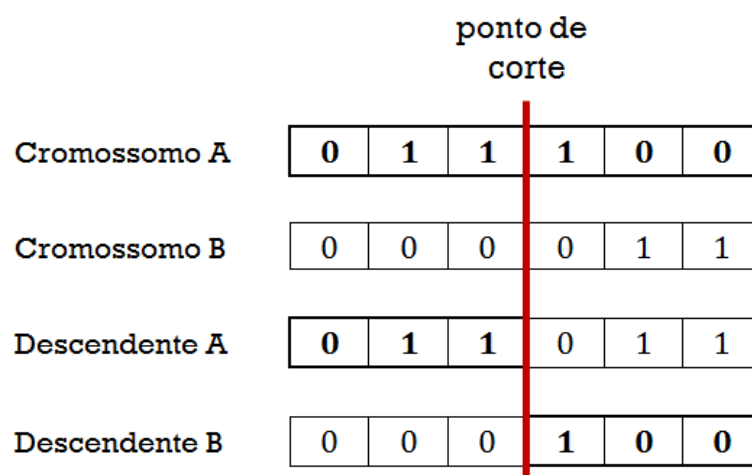


Figura 2.13 – Recombinação entre dois cromossomos

A recombinação pode utilizar dois pontos de seccionamento (figura 2.14). Essa estratégia visa criar mais pontos de cópia dos genes buscando uma maior mistura dos genes dos pais.

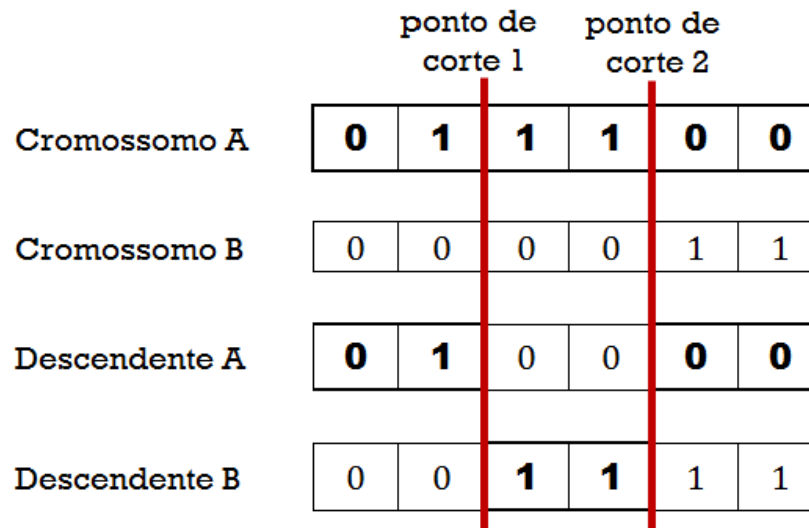


Figura 2.14 – Recombinação entre dois cromossomos, usando dois pontos de corte

2.4.4 Mutação

A mutação é usada para propiciar a renovação do material genético da população. Um ponto importante dentro dos AGs é a taxa de mutação que será usada. Se essa taxa for alta pode ocorrer de o AG tornar-se uma busca aleatória não ocorrendo adaptação.

A mutação do gene ocorre de acordo com o nível de mutação definido. Na codificação por valor, quando ocorre mutação, o gene tem seu valor redimensionado por um percentual aleatório. No exemplo da figura 2.15, o terceiro gene teve seu peso redimensionado por um fator 2.

Cromossomo A	0,5	0,8	1,2	0,1	2,3	1,1
---------------------	-----	-----	-----	-----	-----	-----

Cromossomo A mutado	0,5	0,8	2,4	0,1	2,3	1,1
--------------------------------	-----	-----	-----	-----	-----	-----

Figura 2.15 – Mutação

2.4.5 Algoritmo

O algoritmo de funcionamento do AG segue basicamente os seguintes passos [41]:

Algoritmo 1: Algoritmo genético

1. Início: Gerar uma população aleatória de n cromossomos.
 2. Aptidão: Para cada cromossomo s da população, aplicar uma função de aptidão $f(x)$.
 3. Nova população: Criar uma nova população de cromossomos, repetindo-se os passos abaixo, até que a mesma esteja completa:
 4. Seleção: selecionar de acordo com sua aptidão (melhor aptidão, mais chances de ser selecionado) dois cromossomos para serem os pais.
 5. Recombinação: realizar a recombinação entre os dois cromossomos selecionados anteriormente.
 6. Mutação: alterar os genes dos descendentes da nova geração, de acordo com a probabilidade de mutação.
 7. Aceitação: colocar a nova descendência na nova população.
 8. Substitua: utilizar a nova população gerada para a próxima rodada do algoritmo.
 9. Teste: Se a condição final foi atingida, pare, e retorne a melhor solução da população atual.
 10. Repita: Volte para o passo 2.
-

O algoritmo apresentado acima, é tipicamente repetido diversas vezes, até atingir algum critério de parada pré-estabelecido.

Para a utilização de AGs, é importante primeiramente encontrar uma representação do problema de modo a permitir que o AG consiga aplicar seus operadores genéticos (seleção, recombinação e mutação) nas variáveis de interesse do problema em questão. O primeiro passo é encontrar uma representação do cromossomo, que é uma estrutura de dados geralmente representada por vetores ou cadeias de valores binários, que representam uma possível solução do problema a ser otimizado, sendo em geral o conjunto de parâmetros da função-objetivo cuja resposta será maximizada ou minimizada [52].

Durante a geração da nova população opcionalmente pode-se usar o elitismo, que é uma técnica que preserva a melhor solução, ou seja, os cromossomos com melhores custos, são mantidos de uma geração para outra, para que não se perca a solução, pois é possível que a melhor solução seja encontrada nas primeiras gerações e portanto deve-se mantê-la para acelerar a convergência.

2.5 MODELOS HÍBRIDOS

O gradiente descendente pode ser usado no BP. O gradiente descendente pode ficar preso num mínimo local impossibilitando de encontrar o mínimo global. Um modo de superar a limitação dos algoritmos de treinamento baseados no gradiente descendente é usar AG para realizar o treinamento como uma evolução das conexões sinápticas. Assim o AG pode ser utilizado para uma busca por um conjunto ótimo de pesos globais sem calcular o gradiente descendente.

A abordagem evolucionária de treino de uma RNA consiste nos seguintes passos [61]:

1. Decodificar cada indivíduo da geração atual, num conjunto de pesos sinápticos e construir a RNA correspondente com os pesos.
2. Avaliar cada RNA usando RMSE. A aptidão do indivíduo (cromossomo) é determinado pelo erro. Quanto maior o erro, menor sua aptidão.

3. Selecionar os pais para a recombinação de acordo com suas aptidões.

4. Aplicar operadores de recombinação e mutação gerando descendentes que formarão a próxima população.

Segundo [61], AGs são bastante ineficientes no ajuste fino de uma busca local, sendo entretanto bons em buscas globais. Portanto uma estratégia complementar consiste em usar o AG para uma busca global inicial, efetuando os ajustes de pesos, e depois usar o BP para uma busca local usando o conjunto de pesos pré-ajustados pelo AG.

A integração dos algoritmos genéticos com as redes neurais artificiais (figura 2.16) é vantajoso no sentido de agregar a capacidade de busca global dos AGs [3] junto com o algoritmo de treinamento e ajuste de pesos backpropagation. O AG ajuda a superar os mínimos locais efetuando uma busca mais ampla no espaço de estados, pois ele é menos suscetível de cair num mínimo local, pois não depende do gradiente, em comparação com os algoritmos baseados no gradiente [61]. Desse modo, denomina-se aqui este modelo híbrido de AG-RNA (algoritmos genéticos com redes neurais artificiais).

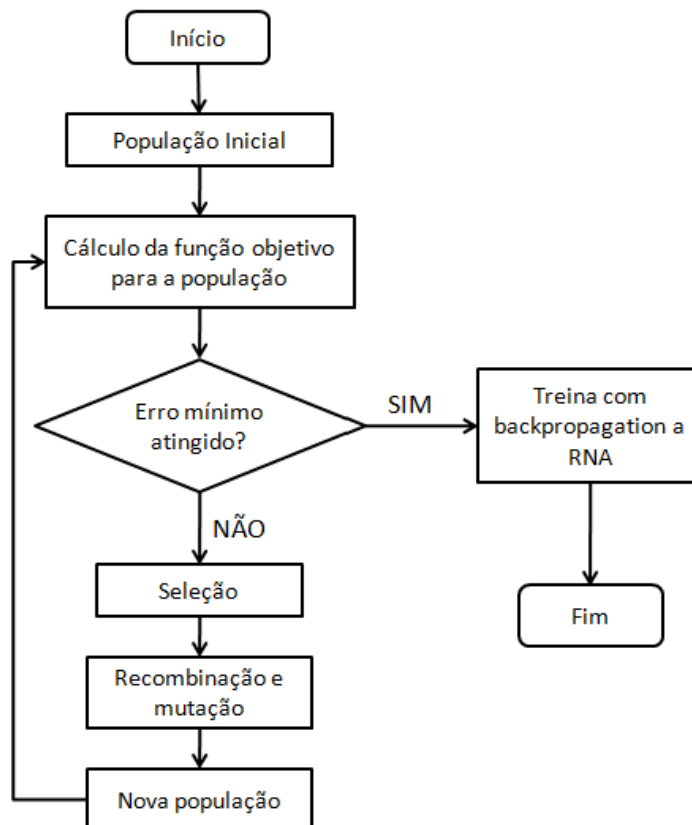


Figura 2.16 – Rede neural, acoplada a um AG

De acordo com a figura 2.16, o primeiro passo do modelo híbrido consiste na formulação de uma população de cromossomos, criados aleatoriamente. Nesse caso, cada gene representa um peso sináptico. Logo em seguida, cada cromossomo é avaliado por uma função de aptidão (função RMSE). Verifica-se na população, se algum candidato obteve aptidão suficiente de acordo com um erro mínimo desejado. Caso contrário, o ciclo do algoritmo genético deve continuar. Para isso seleciona-se os candidatos (método do truncamento), aplica-se recombinação, gerando novos indivíduos, que sofrerão alguma mutação e logo em seguida alocados na nova população. Esse processo repete-se até que o erro mínimo seja atingido ou um número máximo de iterações seja ultrapassada. Após a finalização do AG, o melhor cromossomo é retirado da população e é instanciada uma RNA com os pesos equivalentes aos genes desse indivíduo. Essa RNA é passada para o backpropagation que executará um refinamento dos pesos encontrados. Desse modo pode-se entender a etapa do AG como uma etapa de atribuição inicial dos pesos da RNA.

Além do modelo híbrido algoritmos genéticos-redes neurais artificiais, pode-se citar ainda outras modalidades de sistemas híbridos, utilizando-se RNAs, como por exemplo:

1. RNA MLP e Ant System. No qual utiliza-se de um algoritmo de otimização por Colônia de formigas para determinar os valores otimizados dos parâmetros de constante de momentum, taxa de aprendizagem e o número de neurônios da camada oculta, conforme apresentado em [23].

2. RNAs e sistemas especialistas: onde a saída produzida pela RNA é analisada por uma série de regras codificadas por um sistema especialista, no intuito de tomada de decisão com base no resultado da RNA [28].

3. RNA e modelos ARMA: proposta de um modelo híbrido no qual a RNA é utilizada para capturar a componente não-linear de uma série temporal e o modelo ARMA é utilizado para capturar a componente linear. A combinação dos dois modelos apresentou vantagens em relação a cada modelo isoladamente [53].

3 METODOLOGIA

Este capítulo apresenta a metodologia utilizada neste estudo, que envolve desde a preparação dos dados para processamento até a apresentação dos algoritmos e modelos de arquiteturas de RNAs utilizadas.

3.1 ESTUDO DE CASO

A bacia hidrográfica utilizada como estudo de caso nesta dissertação foi a bacia do rio Piquiri (figuras 3.1 e 3.2), localizada no estado do Paraná, sendo sub-bacia da bacia do rio Paraná.

A rede de estações hidrometeorológicas da bacia do Rio Piquiri é gerenciada pela Superintendência das Águas do Paraná (SUDERHSA).

Neste trabalho foram usados os dados da seguinte estação fluviométrica:

Código ANEEL: 64767000

Nome da estação: Porto Carriel

Município: Laranjal

Área drenagem: 3620 Km²

Latitude: 24° 56' 48''

Longitude: 52° 35' 47'

Altitude: 380 m

Entidade: ANA

Data instalação: 27/05/1981

O Rio Piquiri nasce no Terceiro Planalto, com 485 quilômetros de extensão aproximada. Na margem direita Cantu, Goio-Bang e Goioerê são afluentes do Rio Piquiri; já na margem esquerda, o afluente é o Rio do Cobre. Este rio é um dos principais afluentes 3.2 da margem esquerda do rio Paraná, nascendo na região centro-sul do estado na localidade do Paiquerê, município de Campina do Simão.

A Bacia hidrográfica do Rio Piquiri é a terceira maior em área física

do estado do Paraná. Apresenta uma área total de 24.731km^2 . O Rio Piquiri que atravessa toda a extensão da bacia no sentido sudeste - centro - oeste, desaguardo no rio Paraná. A população total estimada para a Bacia do Piquiri em 2005 foi em 596 mil habitantes, abrangendo o território de 71 municípios.

A bacia do Rio Piquiri é uma das principais do Paraná e está numa das regiões onde o Governo do Estado mantém inúmeras parcerias para assegurar a preservação ambiental repovoamento de rios, recuperação da mata ciliar e parcerias com instituições de ensino, setor produtivo e gestores municipais, por exemplo.

A vazão do Rio Piquiri mais próxima a sua foz, no período entre 1959 e 2005, na estação hidrométrica de Balsa, registrou máxima de $6.258\text{m}^3/\text{s}$ em maio de 1992, mínima de $70,0\text{m}^3/\text{s}$ em dezembro de 1988 e média de $524\text{m}^3/\text{s}$.

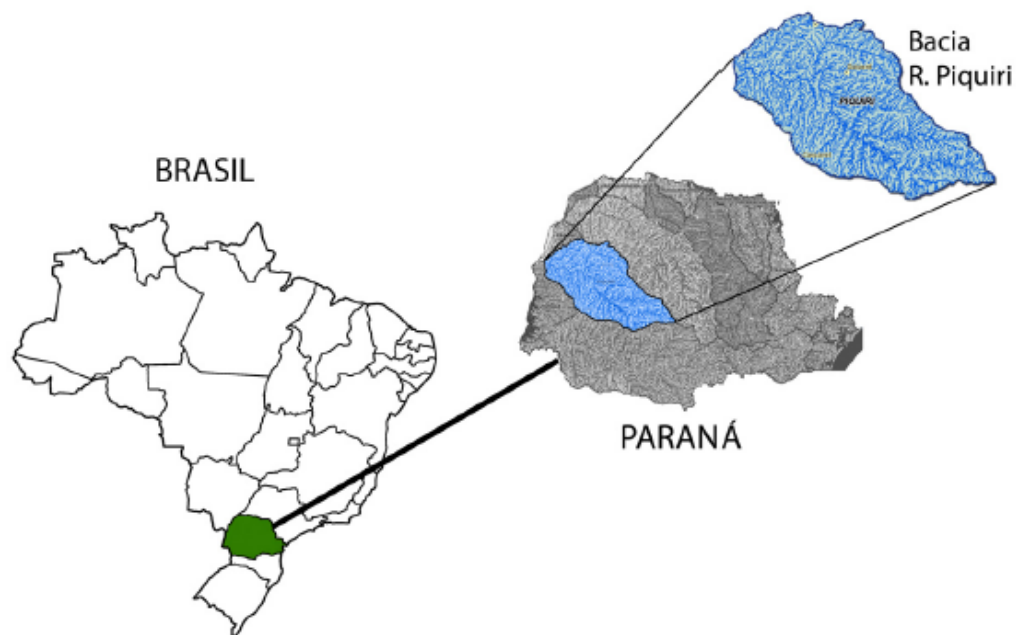


Figura 3.1 – Bacia do rio Piquiri



Figura 3.2 – Bacia do rio Piquiri - Afluentes - adaptado de [43]

Os eventos de vazão utilizados no treinamento, teste e validação da rede neural compreendem um período de aproximadamente 9 anos. Nos hidrogramas abaixo (figuras 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11), tem-se o registro da vazão em escala de medição diária iniciando no ano de 2000 e terminando em 2008, contabilizando um total de 3103 dias de dados de vazão observados. No dia 05.10.2008 (figura 3.11) a vazão apresentou uma máxima de $1991,5 \text{ m}^3/\text{s}$, sendo que no dia 24.07.2006 (figura 3.9) apresentou uma vazão mínima de $6,47 \text{ m}^3/\text{s}$.

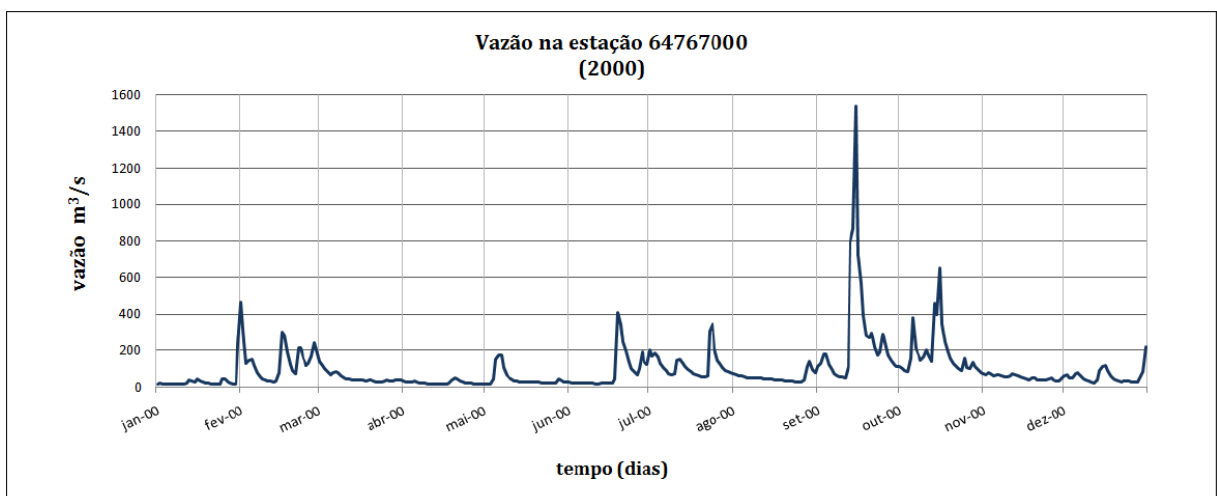


Figura 3.3 – Hidrograma do Rio Piquiri - medições diárias de vazão - Período: 2000

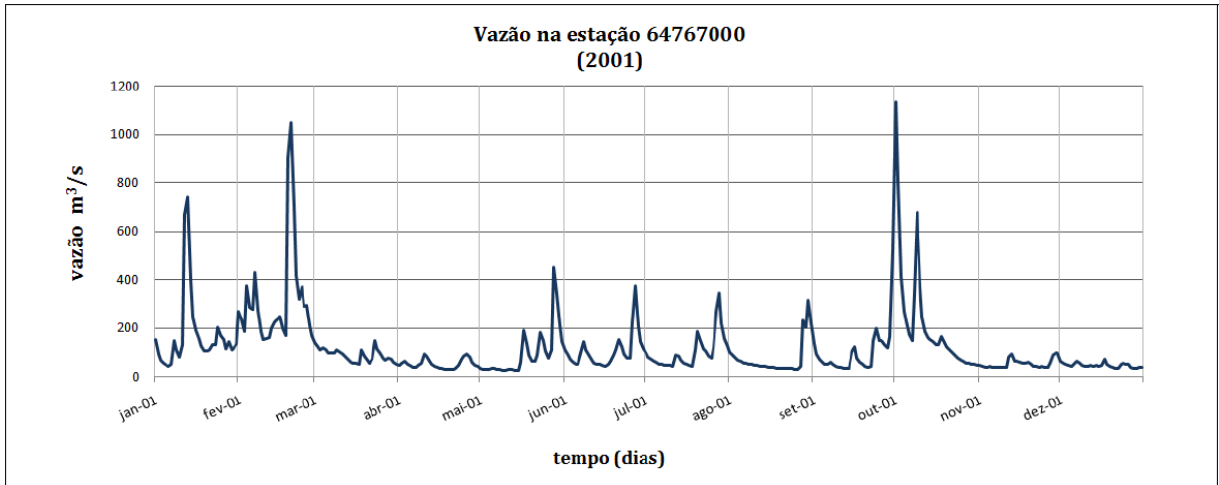


Figura 3.4 – Hidrograma do Rio Piquiri - medições diárias de vazão - Período: 2001

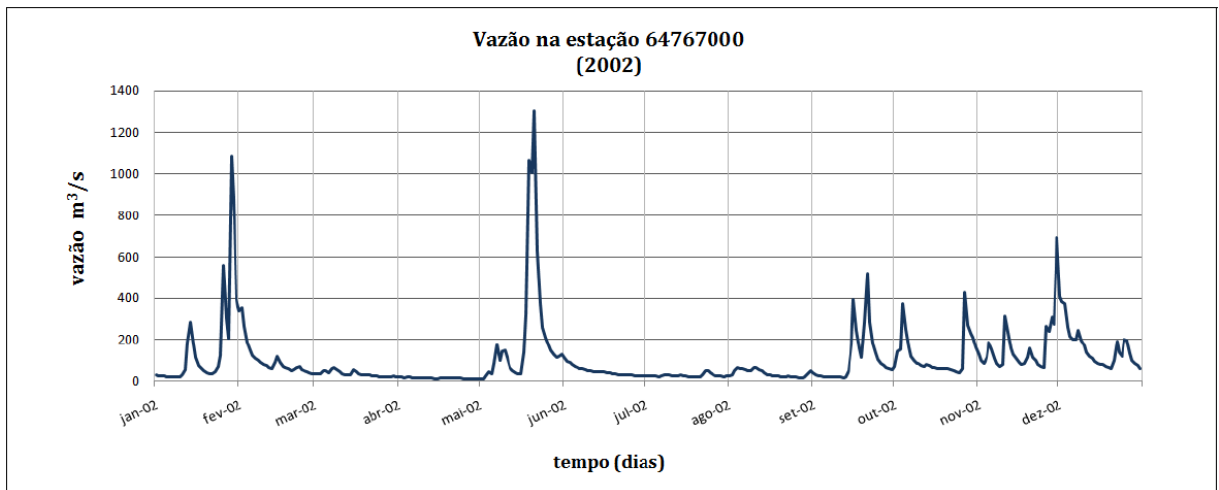


Figura 3.5 – Hidrograma do Rio Piquiri - medições diárias de vazão - Período: 2002

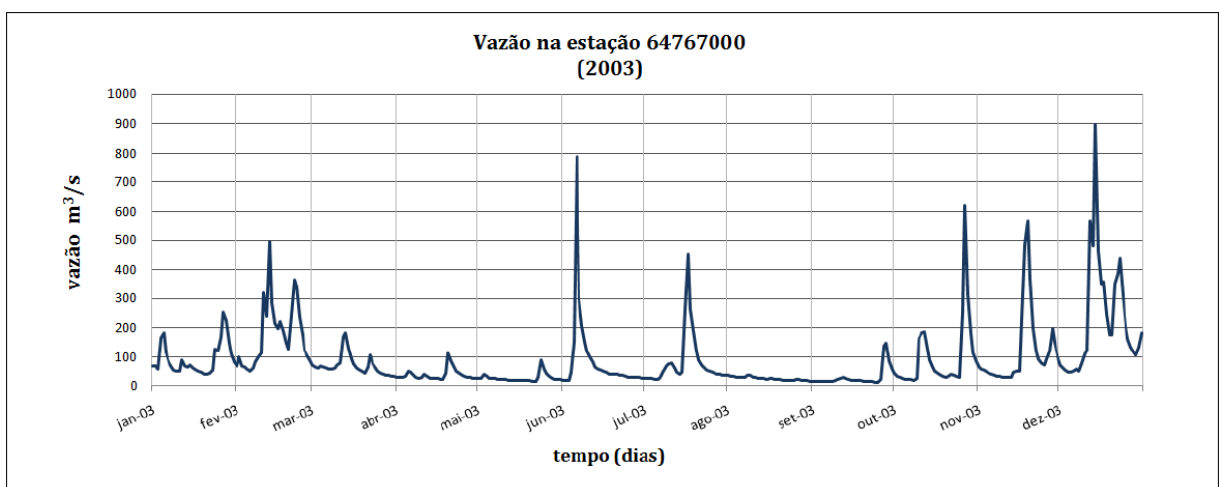


Figura 3.6 – Hidrograma do Rio Piquiri - medições diárias de vazão - Período: 2003

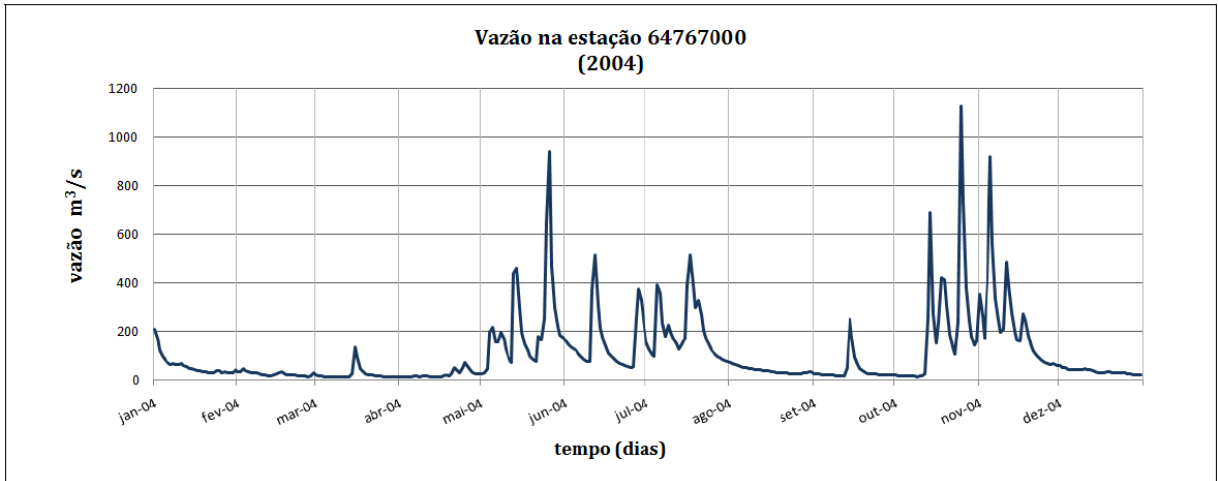


Figura 3.7 – Hidrograma do Rio Piquiri - medições diárias de vazão - Período: 2004

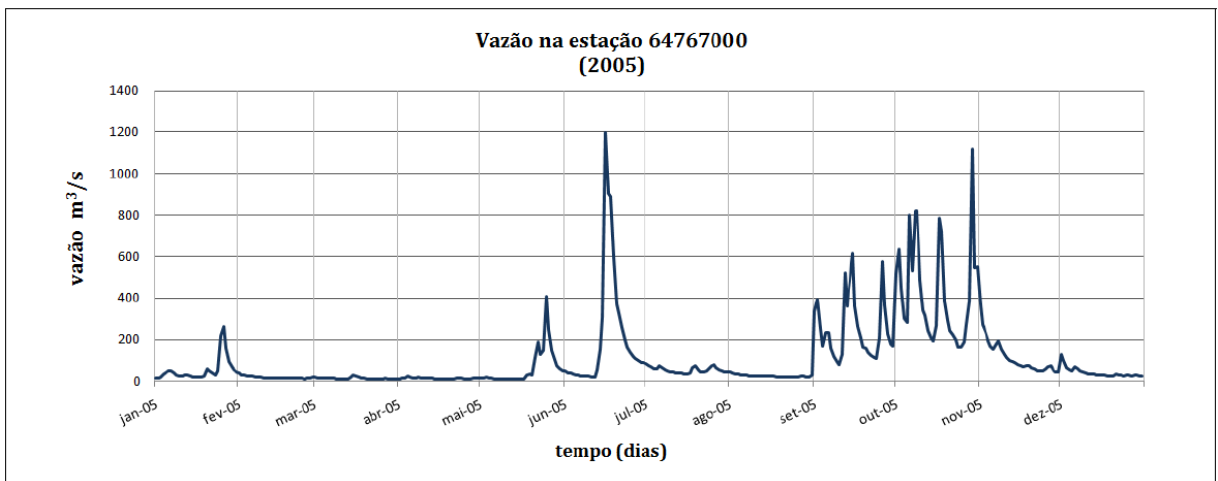


Figura 3.8 – Hidrograma do Rio Piquiri - medições diárias de vazão - Período: 2005

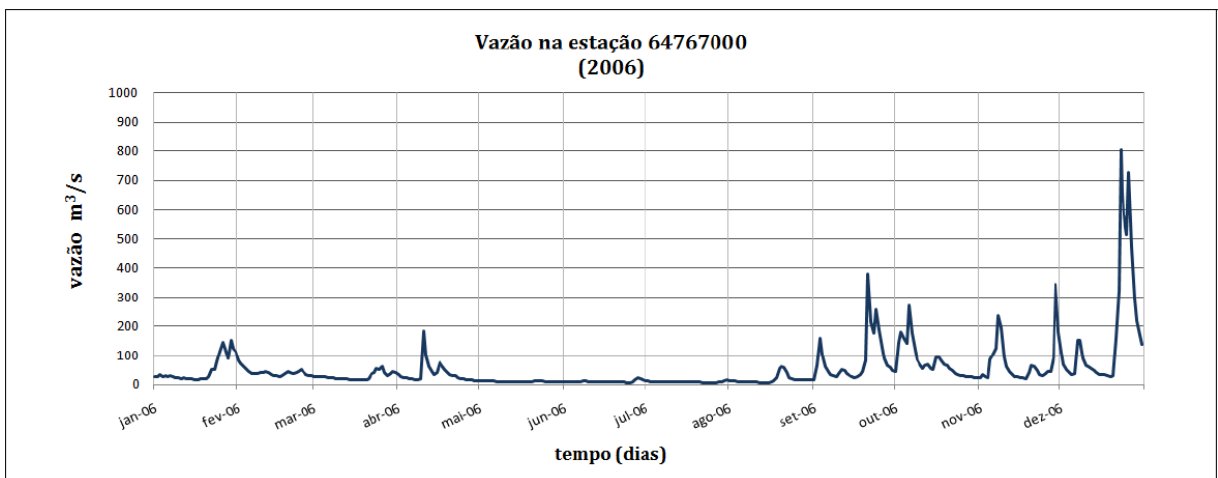


Figura 3.9 – Hidrograma do Rio Piquiri - medições diárias de vazão - Período: 2006

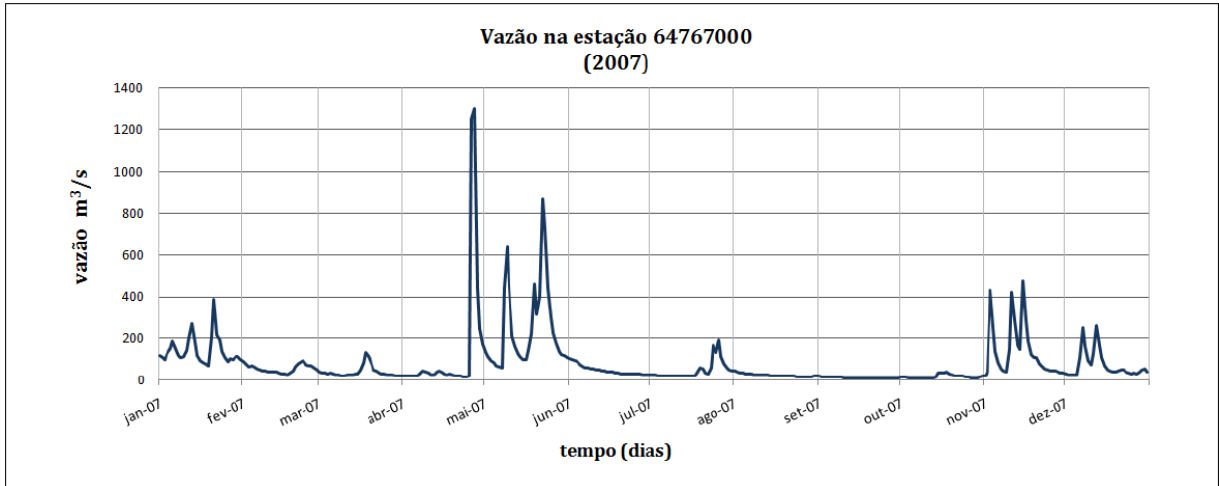


Figura 3.10 – Hidrograma do Rio Piquiri - medições diárias de vazão - Período: 2007

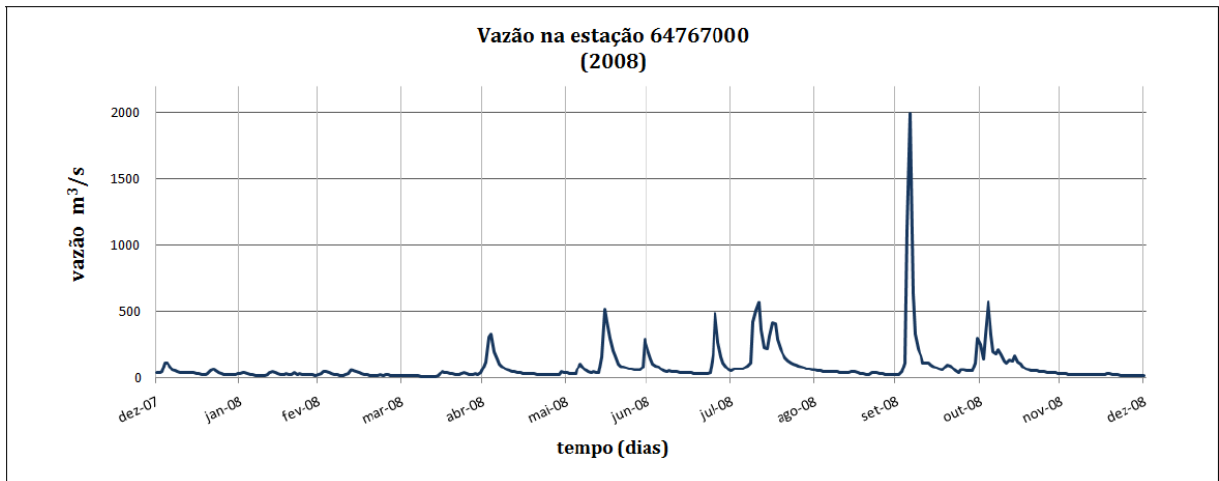


Figura 3.11 – Hidrograma do Rio Piquiri - medições diárias de vazão - Período: 2008

3.1.1 Seleção dos Dados para Processamento

Na bacia do rio Piquiri existem ao todo 67 estações pluviométricas mantidas e monitoradas pela SUDERHSA que foram cedidas para a realização desta pesquisa. Destas, 10 foram descartadas por apresentarem erros ou descontinuidade de medições. Cada estação possui uma série histórica em escala diária de medição, com 3103 registros, na qual cada registro representa o total de chuva acumulado no período de 1 dia.

Após a seleção das séries históricas de precipitação, foi calculado o índice de correlação (equação 3.1) delas em relação a série histórica de vazão (Zq) obtida da estação fluviométrica 64767000. Do resultado obtido, selecionou-se as 5 estações pluviométricas que apresentaram o maior índice de correlação (ver quadro 3.1). Como esperado, estas estações localizavam-se geograficamente nas proximidades da estação de vazão. Estas correlações podem ser visualizados no quadro 3.1.

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (3.1)$$

Tal que X_i é a i -ésima observação da vazão, \bar{X} é a média das observações de vazão, Y_i é a i -ésima observação de precipitação e \bar{Y} é a média das observações de precipitação.

Cod. da Estação Precipitação	Série histórica	Correlação
2552019	Zp_1	0,466
2552008	Zp_2	0,476
2552006	Zp_3	0,417
2452019	Zp_4	0,414
2451010	Zp_5	0,436

Quadro 3.1 - Índices de correlações da estação 64767000 (fluviometria) com as estações de pluviometria

A partir do cálculo da média (equação 3.2) das séries históricas de cada estação pluviométrica, foi gerada uma nova série histórica de precipitação Zp_m , contendo os valores médios diários de chuva. A correlação da série histórica média (Zp_m), em relação a série histórica da vazão (Zq) calculada foi de 0,505 para um período de 3103 dias (ver quadro 3.2). Observa-se que este índice representa uma baixa correlação entre as duas variáveis (chuva-vazão). Isso evidencia a existência de outros fatores que influenciam a vazão, além da precipitação. Poderia-se investigar a causa em termos hidrológicos (exemplo: temperatura, tipo de solo, evapotranspiração, infiltração), entretanto como conhece-se apenas os dados de

chuva e vazão dessa bacia, não é possível proceder com tal análise.

$$Zp_m = \frac{(Zp_1 + Zp_2 + Zp_3 + Zp_4 + Zp_5)}{5} \quad (3.2)$$

Como entrada na rede neural, foi utilizada a série histórica Zp_m e como saída, a série histórica de vazão (Zq). Em alguns modelos, que serão discutidos na seção

3.1.7, é utilizado como entrada as duas séries temporais (Zp_m e Zq).

Estações	64767000	2451010	2452019	2552006	2552008	2552019	Zp_m
Tipo	Vazão(Zq)	Precipitação	Precipitação	Precipitação	Precipitação	Precipitação	Precipitação
Correlação	***	0,436	0,414	0,417	0,476	0,466	0,505
01/01/2000	19,9	0	0	0	0	10,5	2,1
02/01/2000	20,6	0	0	0	0	0	0
03/01/2000	18,15	14,6	0	5,5	0	0	4,02
04/01/2000	16,05	29,8	0	0	0	0	5,96
...
31/12/2008	14,75	3	5,5	0	4,3	0	2,56

Quadro 3.2 - Séries temporais utilizadas

3.1.2 Pré-processamento

Antes de serem submetidos a rede neural, as séries históricas de vazão e chuva foram normalizadas no intervalo de $[0.1 : 0.9]$. Essa etapa é importante para a adequação das variáveis em um mesmo intervalo, pois geralmente elas estão em diferentes escalas de medida como é o caso da precipitação e da vazão. A normalização é feita de acordo com a equação 3.3:

$$X_{norm} = 0.1 + 0.8 \left(\frac{X_i}{X_{max}} \right) \quad (3.3)$$

3.1.3 Divisão dos Conjuntos

Primeiramente o conjunto de dados é dividido em dois subconjuntos: um conjunto de treinamento e outro de teste. O conjunto de treinamento é sub-

dividido em dois outros conjuntos: um subconjunto de estimação e outro de validação. Assim, o subconjunto de estimação é usado para treinar a RNA enquanto que o subconjunto de validação é utilizado para selecionar o melhor modelo de RNA.

O método adotado para treinamento e validação da rede neural foi o método da validação cruzada múltipla (VCM) [38]. Nesta modalidade divide-se o conjunto de treinamento em K subconjuntos, sendo $K > 1$. Em seguida a rede neural é treinada com todos os subconjuntos, com exceção de um deles, que será utilizado para validar a rede medindo-se o RMSE e o R^2 desse conjunto que ficou de fora. O procedimento repete-se K vezes, onde a cada vez um subconjunto diferente é escolhido e usado para coletar as estatísticas de erro de RMSE e R^2 de validação. A avaliação da eficiência do modelo é realizado pela média do RMSE e R^2 obtido em cada uma das K validações.

O tamanho total do conjunto de dados contempla aproximadamente 9 anos. Optou-se em definir para a VCM um $K = 8$. Desse modo, tem-se K subconjuntos com aproximadamente 1 ano de dados de chuva-vazão. Assim, sendo $K = 8$, tem-se 8 anos de treino/validação e o restante dos dados, o equivalente a 1 ano, para teste da rede.

Os passos adotados na divisão dos dados foram os seguintes:

1. Divide-se o conjunto de 9 anos de dados em dois: treinamento e teste.
2. Define-se o subconjunto de treinamento com 8 anos de dados.
3. Define-se o subconjunto teste, com 1 ano de dados.
4. Divide-se o subconjunto treinamento em K subconjuntos, onde $K = 8$:
5. Executa a validação cruzada múltipla no conjunto treinamento.
6. Coleta os dados de validação do modelo (R^2 e RMSE, fazendo a média).
7. Usa a rede para o conjunto de teste (1 ano), que não foi usado na validação.

3.1.4 Método da parada Antecipada

De acordo com [38], o método da parada antecipada consiste na identificação de um momento que seja considerado como um critério de parada do treinamento de uma RNA. Esse ponto de parada é definido de acordo com o acompanhamento do erro de treinamento e do erro de validação. Nas primeiras iterações de treinamento esses dois indicadores tendem a diminuir, sendo que gradativamente, conforme o número de épocas aumenta, a tendência é que esses erros desacelerem. Conforme o número de iterações continua a aumentar, o erro de validação pode aumentar enquanto que o erro de treinamento continua estável ou diminuindo (ver figura 3.12). Essa situação indica que a rede está entrando numa condição de supertreinamento. Assim, esse ponto pode ser usado como um critério de parada do treinamento ao invés do método de definição de um número máximo e fixo de épocas ou de um erro mínimo que são valores que dependem muito do problema em si modelado. Em geral, esse processo pode ser descrito nas seguintes etapas:

1. A cada t períodos o treinamento da RNA é suspenso e a rede passa a operar em sua modalidade direta, apenas calculando a saída. O erro de validação é calculado. Nesse caso, $t == 50$, significa que depois de 50 épocas de treinamento na RNA, o mesmo é suspenso e calculado o erro de validação. Caso o erro seja menor que o anterior, então o treinamento é retomado.
2. A seguir a rede passa a operar em modo treinamento, voltando a ajustar os pesos e os níveis de bias.
3. A evolução do erro de validação é monitorada, e caso seja observado um aumento desse erro de uma iteração n para a iteração $n + 1$, o processo de treinamento é portanto finalizado.

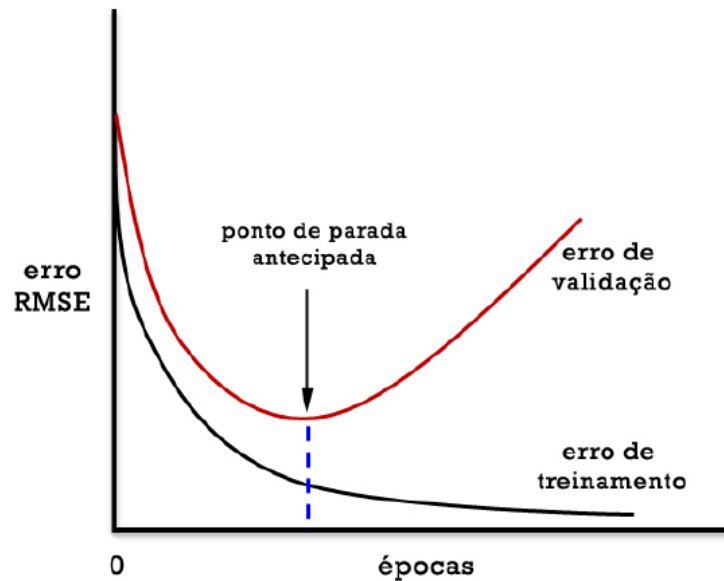


Figura 3.12 – Método da parada antecipada

3.1.5 Definição do número de neurônios da camada oculta

Segundo [25] uma das questões mais importantes e não bem definidas na literatura sobre RNAs é qual modelo de arquitetura utilizar para um determinado problema. Nesse caso, a definição da arquitetura envolve a seleção adequada do número de nodos da camada oculta e do número de neurônios da camada de entrada. Sendo assim, um ponto de partida para a definição do número de neurônios ocultos é a definição de alguma metodologia que permita selecionar apropriadamente um modelo que represente adequadamente o problema em questão. Nesse sentido optou-se pela técnica apresentada por [25] e [59] na qual parte-se de um ponto de vista bottom-up, isto é, inicia-se a rede neural, com um número mínimo de neurônios ocultos, e gradualmente aumenta-se esse número até atingir um critério de parada. Esse procedimento pode ser visualizado na figura 3.13.

A ideia consiste na definição de um modelo inicial de RNA, contendo um número n mínimo de neurônios ocultos, sendo inicialmente $n = 1$. Esse modelo é treinado usando validação cruzada múltipla, calculando o erro médio (RMSE) dos K subconjuntos de validação. Ao término dessa etapa, acrescenta-se um neurônio na camada oculta, repetindo o treinamento e calculando novamente o erro. Se essa nova rede contendo mais neurônios mostrar um erro de validação menor em relação

a rede anterior, então o novo modelo é aceito e parte-se para uma nova hipótese de modelo. Esse processo pára quando unidades adicionais de neurônios ocultos não conseguem reduzir o erro de validação em relação ao modelo anterior. A figura 3.14 exemplifica essa técnica, onde no eixo das abscissas tem-se o número de neurônios na camada oculta e no eixo das ordenadas tem-se o erro RMSE produzido pelo modelo. Observa-se no exemplo, que o menor erro ocorre quando o número de neurônios atinge 3 nodos ocultos. Esse é portanto o ponto escolhido para definição do número de neurônios da camada intermediária.

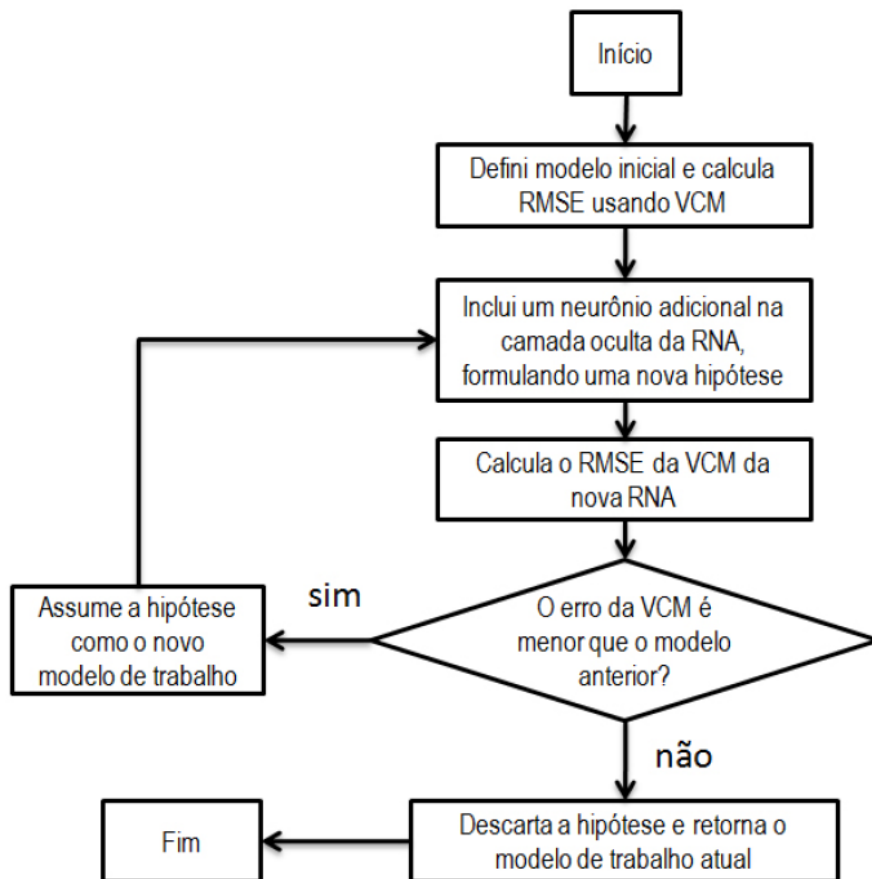


Figura 3.13 – Seleção de um modelo de RNA

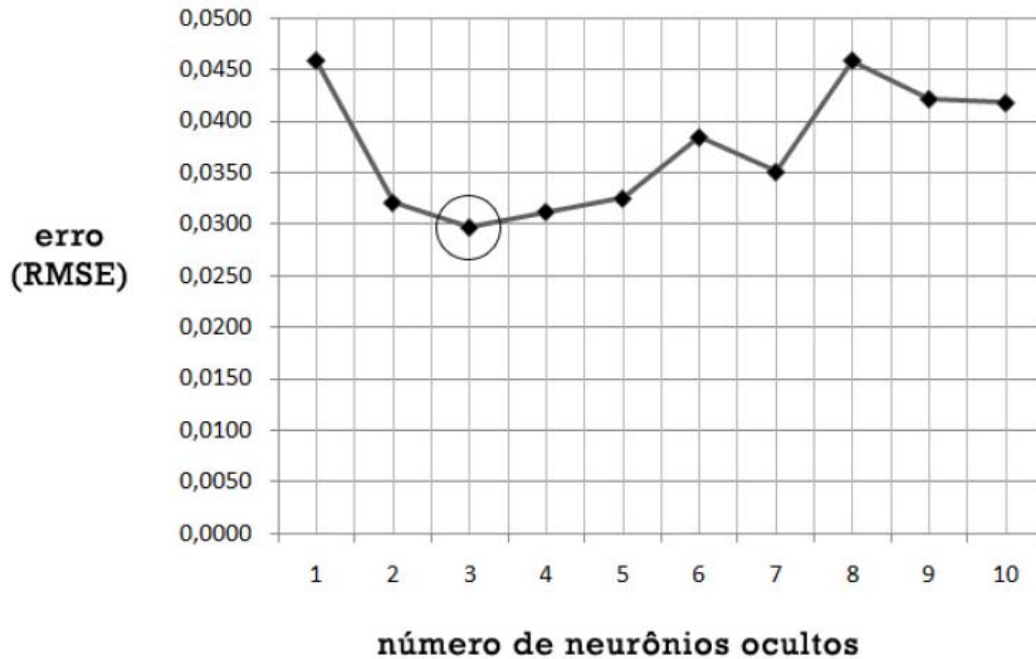


Figura 3.14 – Número de nodos ocultos e os respectivos erros

3.1.6 Definição dos pesos iniciais usando AG

Uma das metodologias testadas neste trabalho envolve o uso de AG acoplada a uma RNA. Nesse caso antes de avaliar cada modelo de RNA, foi utilizado um algoritmo genético para iniciar a matriz de pesos da RNA. A metodologia de integração de algoritmos genéticos com redes neurais inicia-se pela codificação dos pesos de cada rede em uma população de cromossomos. A codificação por valor foi utilizada para representação de cada peso sináptico, em um respectivo gene. Nessa codificação, o gene é codificado como um tipo double, do mesmo tipo do peso sináptico da RNA. O conjunto de genes formam um cromossomo. Cada cromossomo representa uma matriz de pesos de uma possível RNA.

Durante a fase inicial do AG, é instanciada uma população com 1000 cromossomos representando 1000 possíveis configurações de redes neurais. Para cada cromossomo, é aplicada uma função de ajuste para avaliar a qualidade da solução de cada cromossomo. Para isso, é utilizada a função RMSE (equação 4.2).

O AG executa até atingir um erro mínimo ou um número máximo de 100 gerações, com o objetivo de escapar de um mínimo local inicial e de modo a preparar uma matriz inicial de pesos que facilite a convergência do algoritmo backpropagation.

Seleção

O método de seleção utilizado neste trabalho foi a seleção por truncamento. Nesta seleção, a população de cromossomos é ordenada de acordo com a função custo de cada indivíduo, de modo que os primeiros elementos da população são aqueles que possuem o menor custo [57]. A seguir, define-se um limiar T para selecionar uma proporção dos T elementos mais bem adaptados da população para recombinação, onde T pode ser $\frac{1}{2}$ ou $\frac{1}{3}$ por exemplo. Essa proporção de indivíduos selecionados é recombinada $\frac{1}{T}$ vezes para a formação da nova população. A proporção T é na verdade um limiar usado para a seleção dos melhores indivíduos da população. Esse percentual foi definido de modo incremental, iniciando em 0.1 e a cada nova tentativa incrementado por um fator de 0.1. O melhor T encontrado foi de 0.4 para escala mensal e 0.3 para escala diária.

Recombinação

Para realizar a recombinação de dois cromossomos é definido de modo aleatório dois pontos de corte de acordo com a figura 3.15, onde o cromossomo 1 representa uma possível solução, isto é, ele codifica os pesos existentes de uma RNA. Utilizando os operadores genéticos de seleção, recombinação e mutação, o AG busca a formulação de uma população de rede neurais que conforme as gerações evoluem, caminha-se para RNAs mais “bem adaptadas”, ao conjunto de dados.

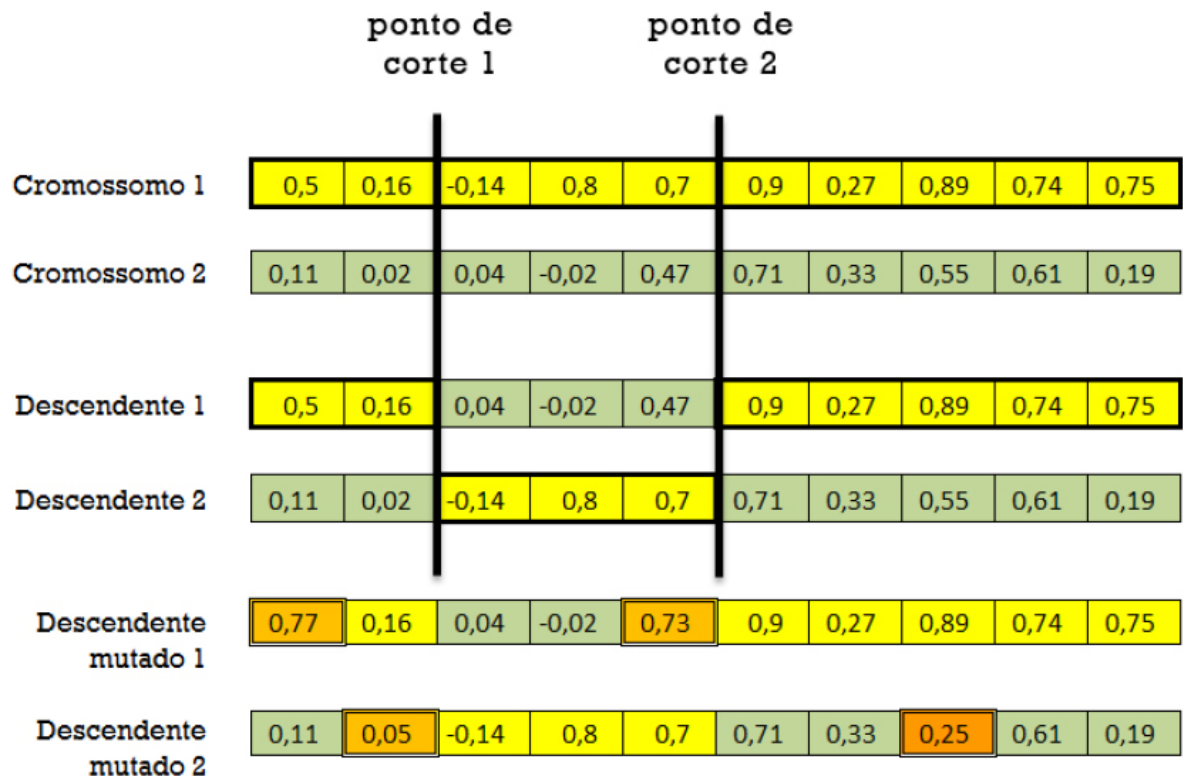


Figura 3.15 – Recombinação com dois pontos de cortes

Em relação a recombinação entre dois cromossomos (figura 3.15), define-se dois pontos de corte, de modo aleatório. A partir da definição do ponto de corte é realizado a recombinação, copiando-se parte do material genético do cromossomo 1 e a outra parte do cromossomo 2. Com isso é gerado sempre dois descendentes para manter-se o mesmo número de indivíduos da população original.

Algoritmo 2: Definição dos pontos de corte

```
ponto1 = random() * (tamanhoGene - quantidadeCorte);
ponto2 = ponto1 + quantidadeCorte;
```

O algoritmo 2 mostra a definição de dois pontos de corte definidos de modo aleatório. O primeiro ponto de corte é definido por uma função aleatória multiplicada pela diferença da quantidade de genes (tamanhoGene) no cromossomo e de um parâmetro que define o quanto de material genético (quantidadeCorte) deve ser cortado quando realizado o cruzamento. O segundo ponto é definido pela soma do primeiro ponto com o quantidadeCorte.

Existem muitas variações de estratégias de pontos de cortes num

cromossomo antes de efetuar a recombinação. Optou-se aqui por utilizar dois pontos de corte, pois de acordo com pesquisa apresentada por [18] APUD DeJong, investigando a eficiência de recombinações com múltiplos pontos de cortes, concluiu que dois pontos de corte demonstraram os melhores resultados, e que aumentando-se esse número, o desempenho do algoritmo genético degradava.

Após a recombinação é aplicada a mutação em um percentual dos genes dos descendentes. Supondo um caso, onde a taxa de mutação seja de 20% e um cromossomo contendo 10 genes, será, nesse caso, escolhido de modo aleatório 2 genes que terão seus valores alterados numa certa proporção, também definida aleatoriamente.

Os valores dos parâmetros do algoritmo genético, tais como população, taxa de mutação e percentual de recombinação foram definidos por um estudo incremental, iniciando-se em valores pequenos e gradativamente aumentando-se os valores até que incrementos não resultassem em melhoria do resultado.

Para taxa de mutação, variou-se o valor iniciando em 0.1 e incrementando-se em 0.1. O melhor valor foi de 0.4, o equivalente a 40% de mutação aplicada nos genes após a recombinação.

A população de indivíduos foi inicialmente estipulada em 100 indivíduos e aumentada até 1000. O melhor valor encontrado, tanto para a série histórica diária quanto a mensal, foi uma população de 1000 indivíduos. Valores acima de 1000, degradam significativamente a performance do AG, devido a grande quantidade de indivíduos que deverão ser recombinados.

3.1.7 Modelagem das RNAs

Em um projeto de aplicação de RNA é preciso especificar diversos parâmetros e fatores relacionados a arquitetura da rede tais como: o número de neurônios em cada camada; número de camadas ocultas; tipo da função de ativação; formato da matriz de entrada dos dados; pré-processamento dos dados e algoritmo de aprendizagem, por exemplo.

Uma rede neural pode ter múltiplas entradas e saídas. Neste

trabalho, a rede usada possui múltiplas entradas e uma saída. As entradas variam de acordo com o modelo especificado no quadro 3.3.3. A saída única representa a vazão do rio em um instante t . Optou-se também pelas redes MLP amplamente avaliadas em outros trabalhos [55],[10],[32],[22], com 3 camadas, sendo 1 camada de entrada, 1 camada oculta e 1 camada de saída. Em [54], [22] foi mostrado que mais de uma camada oculta não exerce fator decisivo no impacto de generalização da rede, podendo inclusive aumentar substancialmente o consumo de processamento [55].

A função de ativação utilizada em cada neurônio artificial foi a função sigmóide. Já o algoritmo de treinamento e ajuste dos pesos sinápticos utilizado foi o backpropagation padrão com constante de momentum. A taxa de aprendizagem e o momentum foram definidos de forma incremental. A taxa de aprendizagem inicial foi definida em 0.05, onde a partir de 0.2 não se observou melhoria no resultado. De modo semelhante, o valor do momentum foi iniciado em 0.1 e gradualmente incrementado até atingir o valor ótimo de 0.9. A taxa de aprendizado foi investigada de modo incremental, sem a utilização de momentum. Depois de obtido a taxa de aprendizagem, foi então investigado a taxa de momentum. Mantendo a taxa de aprendizado fixo, variou-se o momentum no intervalo de [0,1..0,9].

O objetivo dos diferentes tipos de modelos é avaliar a influência de diferentes lags das variáveis de entrada na saída da rede. O modelo mais simples, nesse caso, é o modelo 1, no qual explica a vazão $Q(t)$ de acordo com a precipitação $P(t)$. O modelo com maior número de lags de entrada é o modelo 19, no qual explica a vazão $Q(t)$ em função da precipitação de $t - 5$ e da vazão de $t - 4$, onde t , em escala temporal diária, representa 1 dia de medição.

Para avaliação do vetor de entrada ideal da RNA, foram estipulados 19 modelos de entrada de dados (quadro 3.3). Estes modelos foram propostos de acordo com aqueles definidos em [58] e [10], sendo propostos novos modelos para investigação de diferentes atrasos nas variáveis de precipitação e vazão. Cada modelo, foi utilizado desse modo para definição do número de nodos e do tipo de variáveis da camada de entrada.

Para cada um dos modelos, foi instanciada uma rede neural, cujo número de nodos ocultos, foi definido pelo método descrito na seção

3.1.5.

Assim, cada conjunto de entrada (vetor sinal) e suas respectivas saídas (vazão), tornam-se um evento, sendo que cada evento é apresentado a rede por turnos. Quando todos os eventos são apresentados a rede, o ciclo é repetido até que um critério de parada seja satisfeito. Nesse estudo foi utilizado o critério da parada antecipada, descrito na seção 3.1.4

Modelo	Entradas	Saída
1	$P(t)$	$Q(t)$
2	$P(t), Q(t-1)$	$Q(t)$
3	$P(t), P(t-1)$	$Q(t)$
4	$P(t), P(t-1), Q(t-1)$	$Q(t)$
5	$P(t), P(t-1), P(t-2)$	$Q(t)$
6	$P(t), P(t-1), P(t-2), Q(t-1)$	$Q(t)$
7	$P(t), P(t-1), P(t-2), Q(t-1), Q(t-2)$	$Q(t)$
8	$P(t), P(t-1), P(t-2), P(t-3)$	$Q(t)$
9	$P(t), P(t-1), P(t-2), P(t-3), Q(t-1)$	$Q(t)$
10	$P(t), P(t-1), P(t-2), P(t-3), Q(t-1), Q(t-2)$	$Q(t)$
11	$P(t), P(t-1), P(t-2), P(t-3), P(t-4)$	$Q(t)$
12	$P(t), P(t-1), P(t-2), P(t-3), P(t-4), Q(t-1)$	$Q(t)$
13	$P(t), P(t-1), P(t-2), P(t-3), P(t-4), Q(t-1), Q(t-2)$	$Q(t)$
14	$P(t), P(t-1), P(t-2), P(t-3), P(t-4), P(t-5)$	$Q(t)$
15	$P(t), P(t-1), P(t-2), P(t-3), P(t-4), P(t-5), Q(t-1)$	$Q(t)$
16	$P(t), P(t-1), P(t-2), P(t-3), P(t-4), P(t-5), Q(t-1), Q(t-2)$	$Q(t)$
17	$P(t), P(t-1), P(t-2), P(t-3), P(t-4), P(t-5), Q(t-1), Q(t-2), Q(t-3)$	$Q(t)$
18	$P(t), P(t-1), P(t-2), Q(t-1), Q(t-2), PMA7$	$Q(t)$
19	$P(t), P(t-1), P(t-2), P(t-3), P(t-4), P(t-5), Q(t-1), Q(t-2), Q(t-3), Q(t-4)$	$Q(t)$

Quadro 3.3 - Modelos de entrada de dados da RNA²

3.1.8 Modelo ARMA

Os modelos ARMA foram desenvolvidos em ambiente Matlab. A aplicação inicia-se pelo carregamento das séries temporais em ambiente Java, e

² Onde $Q(t)$ representa a vazão no instante t e $P(t)$ representa a precipitação no instante t . No modelo 18, a entrada PMA7 refere-se a média móvel de 7 dias da precipitação, conforme proposta de [10].

transfere o controle de execução temporariamente para o Matlab, passando juntamente as séries temporais. No Matlab foi desenvolvido um script que testa o modelo ARMA para até 10 atrasos p e q (foi testado para atrasos maiores, mas não houve mudança significativa). Inicia-se fixando $q = 1$ e variando-se p no intervalo [1..10]. Em seguida, q é incrementado ($q = 2$), e assim sucessivamente até $q \leq 10$. Para cada novo valor de q , é testado p no intervalo [1..10]. Para cada modelo produzido, é calculado o AIC correspondente do modelo. Depois de finalizado o processo de variação de q e p , é selecionado o modelo que apresentou o menor AIC. O toolbox utilizado foi o System identification, em especial as funções descritas a seguir:

- **armax()**: função que estima o modelo ARMA.
- **compare()**: função que compara a saída produzida pelo modelo com a saída real.

3.1.9 Ferramentas de Desenvolvimento

O código fonte dos algoritmos utilizados neste trabalho foram implementados na linguagem Java 1.6 e no software Matlab (versão 7.9 - R2009b). Toda parte relacionada as redes neurais e algoritmos genéticos foi desenvolvida no Java utilizando o Encog (v. 2.5)[39], [40], que é uma biblioteca Java de programação de redes neurais. Já a parte relacionada aos modelos ARMA, foi implementada no software Matlab, usando como biblioteca o toolbox System identification [50]. Com o objetivo de integrar as duas plataformas (Java e Matlab) para sistematizar o carregamento de arquivos e análise dos resultados, foi utilizado a biblioteca Matlabcontrol (v. 3.1) [44] que é uma API para controlar e interagir com sessões do software Matlab, a partir de uma aplicação Java. A visão geral da arquitetura do sistema pode ser visualizada na figura 3.16.

Para avaliar os modelos de RNAs foram implementadas diversas classes no java para as seguintes funções: 1. carregamento das séries temporais; 2. preparação das matrizes de entrada de acordo com cada modelo; 3. definição dos modelos de RNAs; 4. funções de cálculo das estatísticas (RMSE e NS); 5. pré-processamento dos dados; 6. preparação dos conjuntos de treinamento/validação de

acordo com a técnica da validação cruzada múltipla (VCM); 7. seleção de modelos de RNAs com identificação do número de neurônios da camada oculta utilizando VCM; 8. integração do Java com o Matlab; 9. treinamento da rede pelo método da parada antecipada.

O equipamento utilizado para rodar as simulações possuía as seguintes configurações: processador intel core i5, 4GB memória DDR3, 500 GB de disco rígido e sistema operacional Windows 7 versão profissional 64 bits.

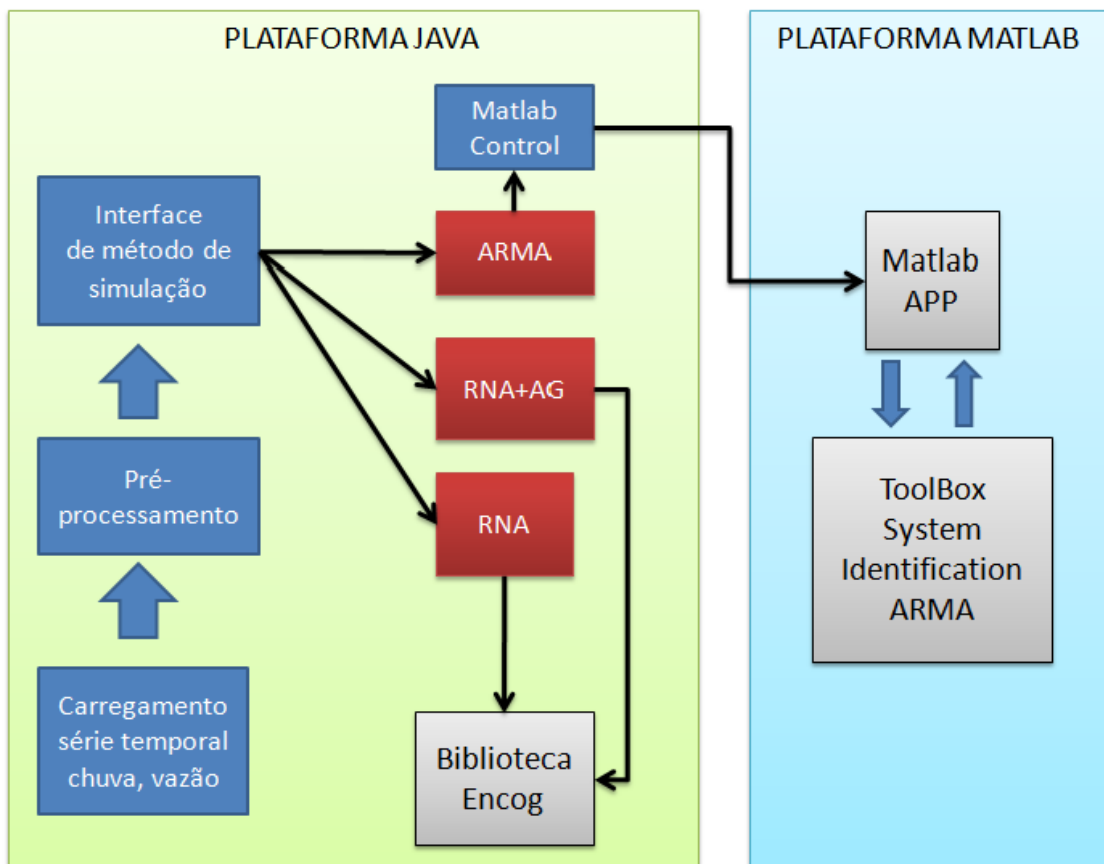


Figura 3.16 – Visão geral da arquitetura do sistema

4 RESULTADOS E DISCUSSÃO

A seguir são apresentados os resultados obtidos com as simulações utilizando redes neurais artificiais, algoritmos genéticos e os modelos autorregressivos de média móvel (ARMA).

4.1 ANÁLISE DOS RESULTADOS

O desempenho de modelos é usualmente avaliado por meio do uso de medidas estatísticas. Para avaliação do desempenho de cada modelo, foram utilizados 2 critérios de avaliação: o índice de eficiência de Nash-Sutcliffe (NS) (equação 4.1) e RMSE (equação 4.2),

$$R^2 = 1 - \frac{\sum_{i=1}^N (Q_i - \hat{Q}_i)^2}{\sum_{i=1}^N (Q_i - \bar{Q}_i)^2} \quad (4.1)$$

onde Q_i é a vazão observada, \hat{Q}_i é a vazão calculada, \bar{Q}_i é a média de vazão e N o número de observações. O índice de eficiência de Nash-Sutcliffe pode variar de $-\infty$ a 1. Um índice igual a 1 corresponde a uma relação perfeita entre a vazão modelada e a vazão observada. Em síntese, quanto mais próximo de 1 o modelo atingir, mais preciso será.

Para avaliação da progressão do erro entre valores observados e previstos, utilizou-se o RMSE (root mean square error):

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Q_i - \bar{Q}_i)^2}{N}} \quad (4.2)$$

onde N é o número total de observações, Q_i é a vazão observada e \bar{Q}_i é a vazão calculada. Quanto menor o erro ou mais próximo de zero, melhor será o ajuste dos dados ao modelo obtido. O RMSE é uma medida frequentemente

utilizada para medir diferenças entre valores previstos por um modelo ou estimados com os valores reais observados. RMSE é uma boa medida de precisão, onde cada diferença individual é também chamada de resíduo e o RMSE serve para unir estes resíduos dentro de uma única medida denotando o poder de previsibilidade do modelo.

4.1.1 Análise preliminar

A partir de um conjunto inicial de 19 modelos (quadro 3.3), selecionou-se aqueles que apresentaram R^2 superior a 0.5 durante uma etapa preliminar de avaliação dos modelos. De fato, os modelos de entrada de dados selecionados para as simulações foram aqueles que utilizavam além da precipitação, também a vazão como entrada na RNA, ou seja, descartou-se os modelos que possuíam apenas a precipitação como dado de entrada, que nesse caso foram os modelos 1, 3, 5, 8, 11 e 14. Esses modelos apresentaram resultados inferiores a 0,5 em termos de R^2 . Em seguida, os modelos com resultado superior a 0,5 foram aplicados em 4 estudos principais que foram os seguintes:

1. Modelagem chuva-vazão usando RNA com AG em escala diária.
2. Modelagem chuva-vazão usando RNA em escala diária.
3. Modelagem chuva-vazão usando RNA com AG em escala mensal.
4. Modelagem chuva-vazão usando RNA em escala mensal.

4.1.2 Modelos AG-RNA em escala diária

No quadro 4.1 tem-se os parâmetros utilizados nas simulações dos modelos, que utilizam algoritmos genéticos em escala diária, sendo: K o número de divisões do conjunto treinamento para a validação cruzada múltipla; N o tamanho da série temporal de chuva-vazão; conjunto treino refere-se aos dados usados para treino e validação dos modelos, sendo nesse caso correspondente a

aproximadamente 8 anos; conjunto teste refere-se a parcela dos dados no qual a rede neural não foi treinada, sendo usado aqui como indicador da capacidade da RNA em prever eventos para os quais não “viu” anteriormente, compreendendo um período de 1 ano; população refere-se a quantidade de cromossomos codificados, nesse caso representando 1000 possíveis RNAs; taxa de mutação e percentual de recombinação são percentuais utilizados para a definição do número de genes que serão modificados (40%) durante a execução do AG e o percentual de cromossomos que serão selecionados para recombinação (30%).

O resultado do primeiro estudo (modelagem chuva-vazão usando RNA com AG em escala diária) é apresentado no quadro 4.2, onde tem-se a arquitetura de rede usada, o número de neurônios da camada oculta, o erro RMSE medido na fase de validação, o coeficiente R^2 também obtido da fase de validação.

Taxa de aprendizado	0.02
Momentum	0.5
K	8
N	3103
Conjunto treino	2740
Conjunto teste	363
População	1000
Taxa de mutação	0.4
Percentual de recombinação	0.3

Quadro 4.1 - Parâmetros utilizados nas simulações AG-RNA-Diário

Conforme observa-se no quadro 4.2, o modelo selecionado foi aquele que apresentou o menor erro durante a etapa de validação cruzada múltipla, sendo portanto nesse caso o modelo 9, que apresentou um erro de 0,02571, utilizando 5 neurônios na camada oculta e 5 neurônios na camada de entrada, obtendo um R^2 de 0,870.

Escala Diária				
Modelo	Arquitetura	Nodos Ocultos	RMSE (validação)	R^2 (validação)
19	10-3-1	3	0,03107	0,859
18	6-3-1	4	0,02912	0,870
17	9-3-1	3	0,02937	0,865
16	8-3-1	3	0,02959	0,862
15	7-2-1	2	0,02910	0,869
13	7-3-1	3	0,02976	0,870
12	6-2-1	2	0,03000	0,871
10	6-2-1	2	0,03001	0,871
9	5-5-1	5	0,02571	0,870
7	5-3-1	3	0,03037	0,867
6	4-2-1	2	0,02975	0,872
4	3-4-1	4	0,02938	0,871
2	2-2-1	2	0,04480	0,714

Quadro 4.2 - Desempenho dos modelos usando AG-RNA

4.1.3 Modelos RNA em escala diária

No quadro 4.3, apresenta-se os resultados obtidos das simulações utilizando uma rede MLP com *backpropagation*, sem o uso de algoritmos genéticos. Observa-se que os resultados da RNA foram próximos daqueles utilizando AG-RNA. Nesse caso, o melhor resultado em termos de RMSE, foi o modelo 4, utilizando 3 entradas e 4 neurônios ocultos, com um erro de 0,02891. Este mesmo modelo apresentou um coeficiente R^2 de 0,873 na fase de validação da rede.

Escala Diária				
Modelo	Arquitetura	Nodos Ocultos	RMSE (validação)	R^2 (validação)
19	10-2-1	2	0,02909	0,868
18	6-2-1	2	0,02921	0,869
17	9-3-1	3	0,02897	0,869
16	8-2-1	2	0,02909	0,869
15	7-2-1	2	0,02989	0,862
13	7-4-1	4	0,02920	0,868
12	6-4-1	4	0,02893	0,870
10	6-3-1	3	0,02910	0,869
9	5-4-1	4	0,02896	0,871
7	5-3-1	3	0,02909	0,870
6	4-3-1	3	0,02900	0,871
4	3-4-1	4	0,02891	0,873
2	2-2-1	2	0,04522	0,701

Quadro 4.3 - Desempenho dos modelos usando RNA

4.1.4 Modelos AG-RNA em escala mensal

Nas simulações em escala mensal não foram utilizados os modelos 19, 18, 17, 16, 15 e 13, devido a restrição do tamanho do conjunto de dados em escala mensal para treinamento e teste. Desse modo utilizou-se somente os modelos 2, 4, 6, 7, 9, 10 e 12 para as simulações mensais pois os modelos de mais alta ordem exigem maior quantidade de variáveis de entrada o que não permite um bom teste do modelo quando existe apenas 14 meses de dados para teste, conforme especificado no quadro 4.4. Neste quadro constam também os parâmetros usados na configuração dos modelos de RNA e AG-RNA em escala mensal. Os valores dos parâmetros do algoritmo genético, tais como População, taxa de mutação e percentual de recombinação e da RNA foram obtidos de acordo com a metodologia discutida nas seções 3.1.6 e 3.1.7.

Taxa de aprendizado	0.2
Momentum	0.9
K	8
N	102
Conjunto treino	88
Conjunto teste	14
População	1000
Taxa de mutação	0.3
Percentual de recombinação	0.4

Quadro 4.4 - Parâmetros utilizados na simulação AG-RNA-Mensal

O quadro 4.5, mostra o resultado das simulações de cada modelo. Nesse estudo, o modelo 6 apresentou os melhores resultados, com 7 nodos ocultos e 4 nodos na camada de entrada, apresentando RMSE de 0,05717 e R^2 de 0,666 na etapa de validação.

Escala Mensal				
Modelo	Arquitetura	Nodos Ocultos	RMSE (validação)	R^2 (validação)
12	6-3-1	3	0,06927	0,666
10	6-5-1	5	0,06874	0,661
9	5-2-1	2	0,07128	0,627
7	5-1-1	1	0,07774	0,599
6	4-7-1	7	0,05717	0,666
4	3-4-1	4	0,07871	0,694
2	2-7-1	7	0,07273	0,653

Quadro 4.5 - Desempenho dos modelos usando AG-RNA

4.1.5 Modelos RNA em escala mensal

O quadro 4.6, mostra o resultado das simulações de cada modelo em escala mensal, usando somente RNA. Nesse estudo o modelo 12 foi o melhor, com 5 nodos ocultos e 6 nodos na camada de entrada, apresentando um RMSE de 0,04767 e R^2 de 0,723 na etapa de validação.

Escala Mensal				
Modelo	Arquitetura	Nodos Ocultos	RMSE (validação)	R^2 (validação)
12	6-5-1	5	0,04767	0,723
10	6-4-1	4	0,05484	0,679
9	5-3-1	3	0,06067	0,711
7	5-5-1	5	0,06078	0,696
6	4-3-1	3	0,06465	0,665
4	3-6-1	6	0,06893	0,659
2	2-2-1	2	0,07531	0,655

Quadro 4.6 - Desempenho dos modelos usando RNA

4.1.6 Comparação das técnicas - Fase de validação

Na etapa de validação dos modelos, a RNA apresentou melhores resultados em termos de R^2 e RMSE quando comparado ao modelo híbrido AG-RNA, conforme apresentado no quadro 4.7:

Técnica	Modelo	Escala	Arquitetura	RMSE (treino)	R^2 (treino)
AG-RNA	9	Diário	5-5-1	0,02571	0,870
RNA	4	Diário	3-4-1	0,02891	0,873
AG-RNA	6	Mensal	4-7-1	0,05717	0,666
RNA	12	Mensal	6-5-1	0,04767	0,723

Quadro 4.7 - Comparação entre RNA e AG-RNA

4.1.7 Fase de testes

A seguir são apresentados os resultados da avaliação da fase de teste dos melhores modelos, selecionados na etapa de treinamento (quadro 4.7). Nessa fase, é apresentado um conjunto de dados reservados, que não foram apresentados anteriormente para a RNA, e portanto, será usado nesse caso para verificação da capacidade de generalização de cada modelo.

Modelos AG-RNA em escala diária

Usando AG-RNA em escala diária, o modelo 9 apresentou um R^2 de 0,856 (figura 4.2) e RMSE de 0,01417, apresentando em alguns pontos da aproximação, um erro principalmente nos picos de vazão, conforme figura 4.1, e boa representação nos períodos de baixa vazão. O gráfico da figura 4.2 mostra um erro maior da previsão do modelo, nas vazões mais altas. Nesse caso, o modelo previu vazões acima daquelas observadas.

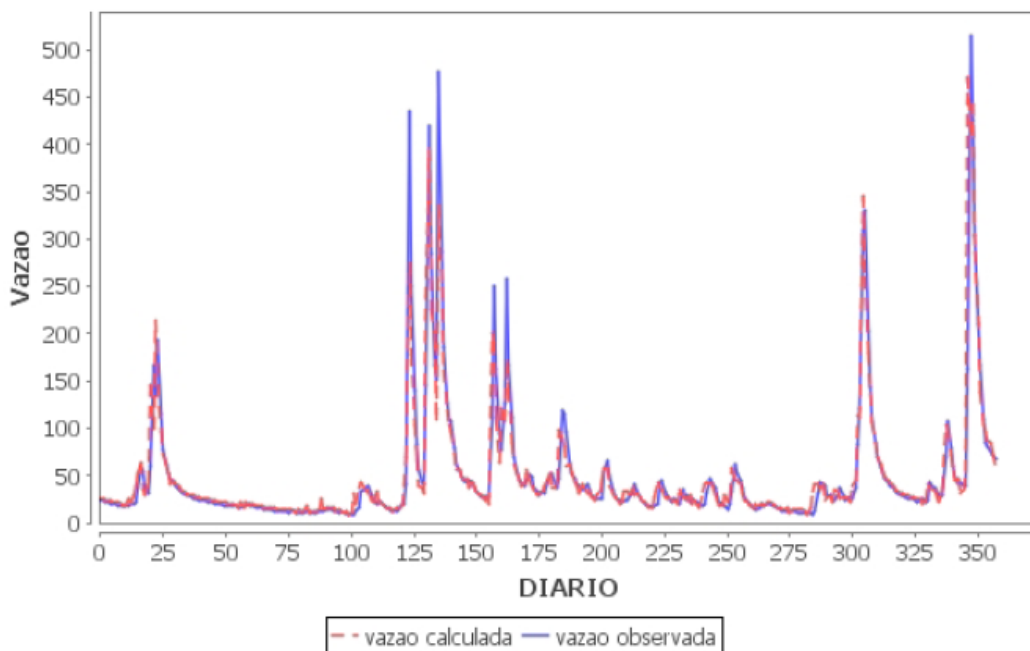


Figura 4.1 – Comparação entre vazão observada e calculada usando AG-RNA

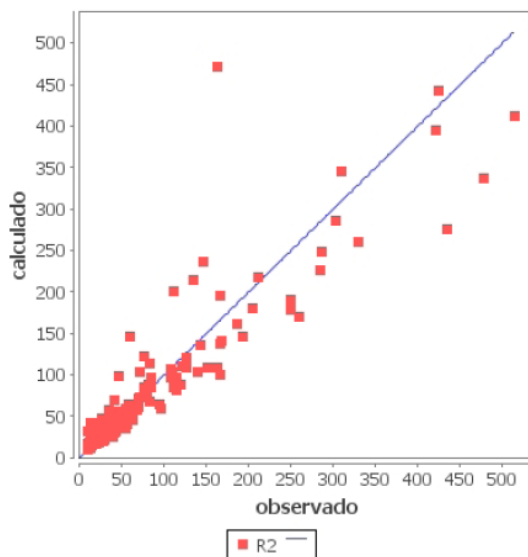


Figura 4.2 - Coeficiente R^2 - AG-RNA

Modelos RNA em escala diária

O melhor modelo RNA obtido em escala diária foi o modelo 4, que apresentou R^2 de 0,811 (figura 4.4) e RMSE de 0,04724. A figura 4.3 mostra um período de teste de 1 ano de previsão (vazão calculada) feita pelo modelo, em comparação com os dados observados (vazão observada). Nesse modelo houve uma superestimação maior dos picos de vazão em comparação ao modelo AG-RNA.

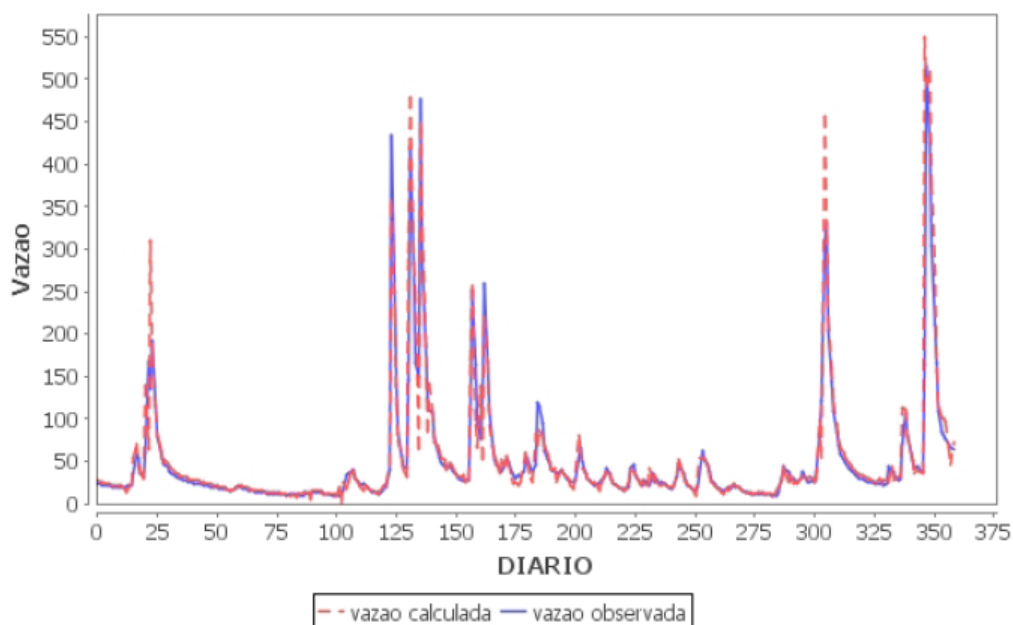


Figura 4.3 – Comparação entre vazão observada e vazão calculada (RNA)

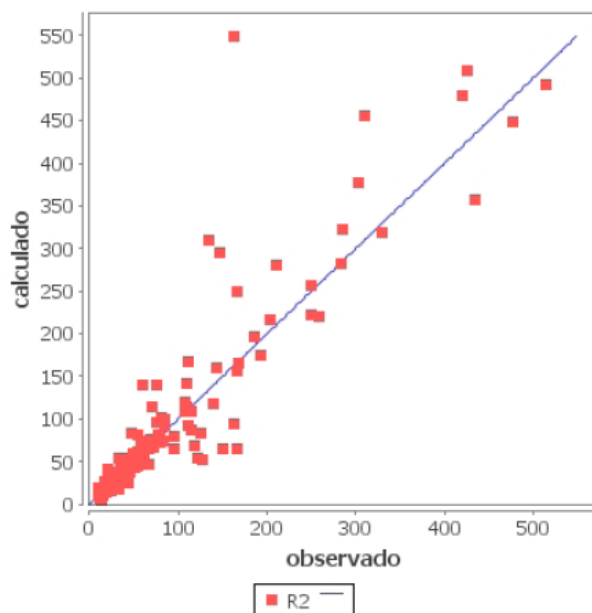


Figura 4.4 – Coeficiente R^2 - RNA

Modelos AG-RNA em escala Mensal

O melhor modelo AG-RNA obtido em escala mensal foi o modelo 6, que apresentou R^2 de 0,888 (figura 4.6) e RMSE de 0,03792. A figura 4.5 mostra um período de teste de 10 meses previstos (vazão calculada) pelo modelo, em comparação com os dados observados (vazão observada).

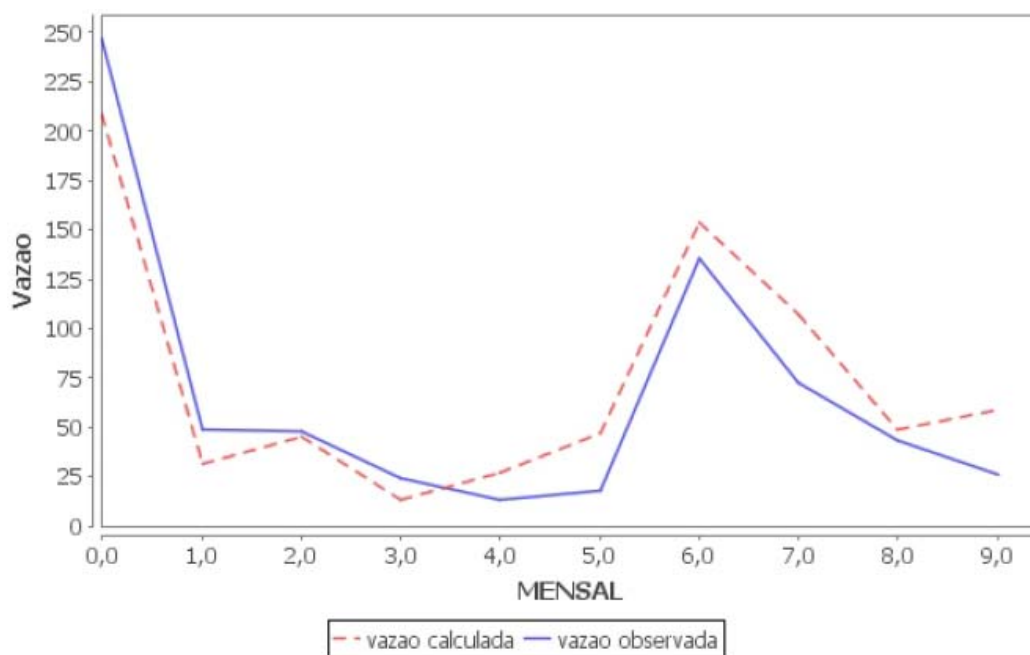


Figura 4.5 – Comparação entre vazão observada e calculada usando AG-RNA

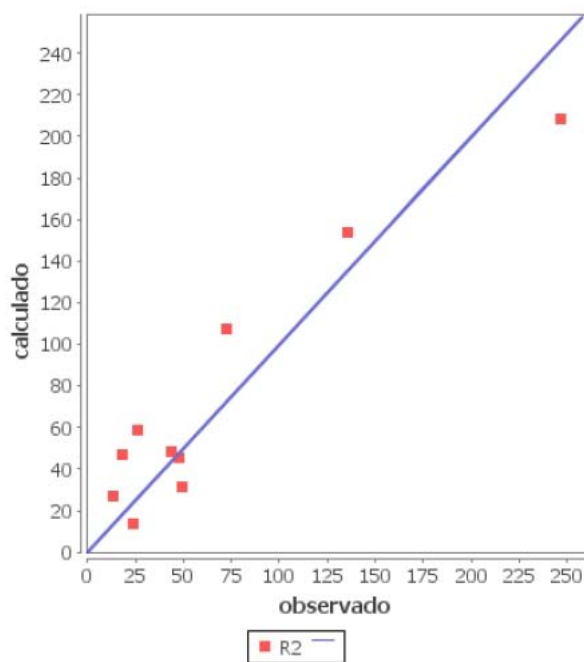


Figura 4.6 – Coeficiente R^2 - AG-RNA

Modelos RNA em escala mensal

O melhor modelo RNA obtido em escala mensal foi o modelo 12, que apresentou R^2 de 0,78 (figura 4.8) e RMSE de 0,11304. A figura 4.7 mostra um período de teste de 8 meses previstos (vazão calculada) pelo modelo, em comparação com os dados observados (vazão observada).

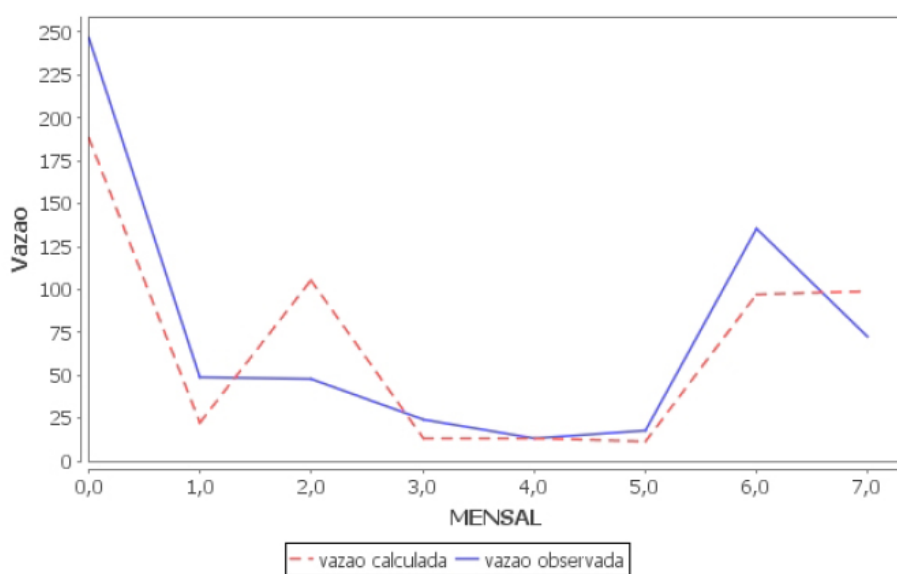


Figura 4.7 – Comparação entre vazão observada e calculada usando RNA

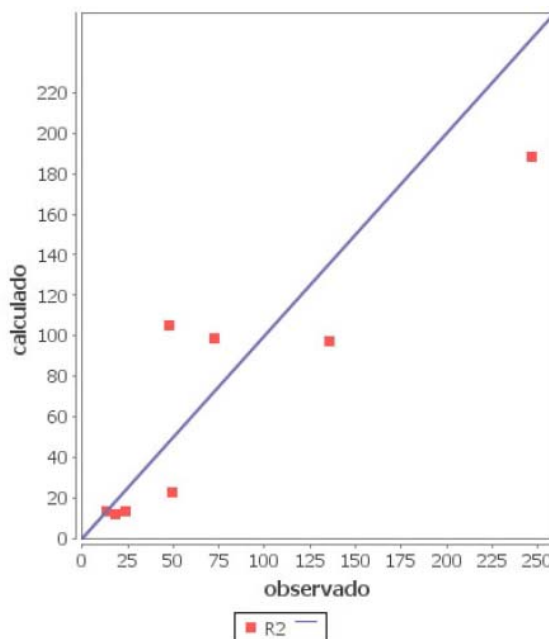


Figura 4.8 – Coeficiente R^2 - RNA

Modelos ARMA em escala diária

O melhor modelo ARMA encontrado, de acordo com o critério AIC (akaike information criteria) [13] [46], foi aquele com $p = 10$, e $q = 5$, conforme apresentado na quadro 4.8:

Modelo	AIC	R^2
ARMA(5,10)	8.857	0.5979
ARMA(6,5)	8.860	0.5950
ARMA(2,2)	8.865	0.5879
ARMA(1,1)	8.914	0.5655

Quadro 4.8 - Resultados obtidos

Na figura 4.9 tem-se um período de 1 ano de teste onde a linha tracejada representa a previsão realizada pelo modelo, comparada com a linha contínua que representa os dados observados utilizados para teste. A figura 4.10 mostra o R^2 de 0,5979 obtido pelo melhor modelo encontrado. Observa-se que o modelo apresentou erros tanto nas vazões baixas quanto nos picos de vazão. Esse modelo utilizou apenas a informação da vazão como entrada.

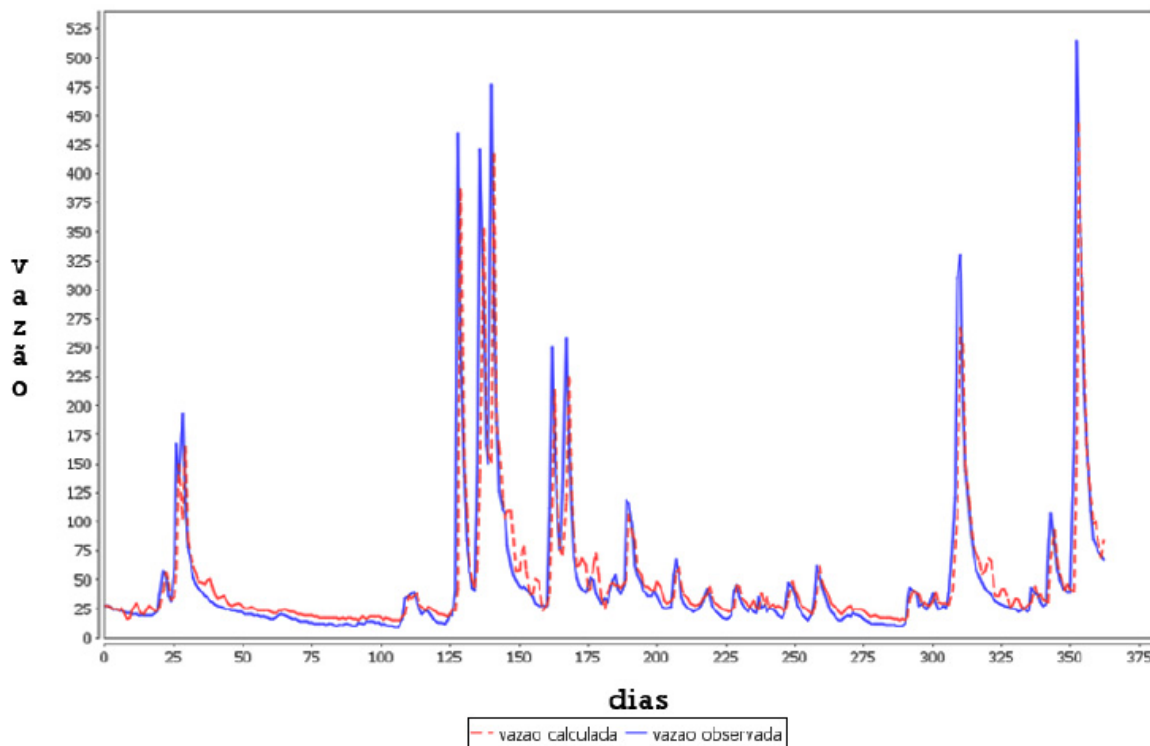


Figura 4.9 – Vazão prevista pelo modelo Arma(10,5)

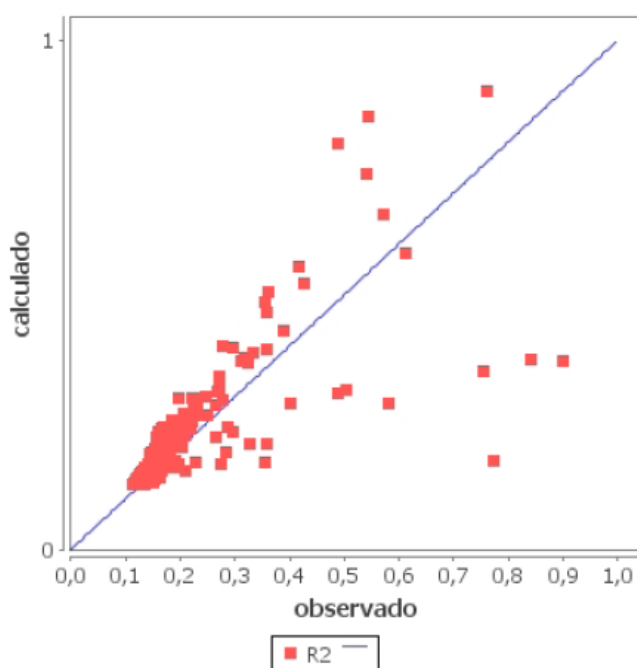


Figura 4.10 – Coeficiente R^2 - modelo Arma(10,5)

Foi testado também o modelo ARMA utilizando-se a vazão e a precipitação como entrada de dados. O melhor modelo obtido nesse caso foi Arma(2,2) com R^2 de 0,56. A série temporal de precipitação é uma série cujos dados

de medição, são altamente não lineares e esse fator influenciou na capacidade de representação do modelo ARMA, mostrando um desempenho pior em relação ao modelo ARMA que usou apenas a vazão como entrada. Isso pode ser devido a vazão apresentar maior suavidade nas oscilações entre as medições vizinhas.

Resultados comparativos

Na etapa de teste da rede, o modelo híbrido AG-RNA apresentou melhores resultados de convergência quando comparado ao de uma RNA, sem AG, conforme apresentado no quadro 4.9, ou seja, os modelos AG-RNA conseguiram uma melhor generalização dos dados em relação aos modelos RNA.

Técnica	Modelo	Escala	Arquitetura	RMSE (teste)	R^2 (teste)
AG-RNA	9	Diário	5-5-1	0,01417	0,856
RNA	4	Diário	3-4-1	0,04724	0,811
AG-RNA	6	Mensal	4-7-1	0,03683	0,888
RNA	12	Mensal	6-5-1	0,11304	0,78

Quadro 4.9 - Comparação entre RNA e AG-RNA

A principal vantagem no uso do AG-RNA está na rapidez de convergência. Em diversas simulações o modelo RNA não conseguia convergir, devido a uma má atribuição da matriz de pesos ficando preso num mínimo local. Isso exigia reinicialização da matriz de pesos, para busca num ponto diferente do espaço de estados. Em algumas situações essa matriz precisou ser reiniciada 4 vezes. O AG-RNA precisou de apenas 1 inicialização para conseguir convergência. Observe também que existe uma correspondência entre o erro apresentado na fase de treino e o erro apresentado na fase de teste. Isso significa que a rede está conseguindo uma boa generalização [29].

Comparando-se os resultados do modelo autorregressivo de média móvel (ARMA) com a RNA, percebe-se que a mesma foi muito superior na representação da modelagem chuva-vazão. O melhor modelo ARMA (5,10) obteve um coeficiente R^2 de 0,5979 (quadro 4.8) enquanto que o melhor modelo AG-RNA obteve um R^2 de 0,836 (quadro 4.9).

De um modo geral, nos modelos em escala diária, é possível observar que as RNAs e os modelos ARMA não conseguiram representar adequadamente os picos de vazão, de modo semelhante ao relatado por [22].

5 CONCLUSÕES E RECOMENDAÇÕES

A vantagem do uso de RNAs na modelagem chuva-vazão está principalmente ligada à redução da dimensionalidade do fenômeno a duas ou mais variáveis. Nesse trabalho, utilizou-se apenas a precipitação e a vazão para treinamento da rede. Os resultados obtidos foram considerados bons, levando-se em consideração a baixa correlação entre os eventos chuva-vazão da bacia estudada.

Um grande número de trabalhos usando RNAs na modelagem chuva-vazão utiliza como entrada na rede, além da precipitação, a temperatura e/ou a evapotranspiração e a vazão. Neste trabalho, foi simplificado ainda mais essa relação, usando-se nesse caso, apenas precipitação e vazão. Nesse cenário a RNA acoplada a um AG, conseguiu realizar uma boa aproximação, sendo que no melhor modelo obteve um coeficiente R^2 de 0.856 em escala diária e 0,888 em escala mensal. Esse resultado foi superior em relação a modelagem utilizando somente RNA, que apresentou um coeficiente R^2 de 0.811 em escala diária e 0,78 em escala mensal. Os resultados obtidos foram considerados bons quando se analisa a baixa correlação entre as duas séries temporais chuva-vazão, no caso de 0,505 e também quando se considera a área total da bacia hidrográfica utilizada como elemento de estudo de caso.

O resultado comparativo entre RNAs e os modelos autorregressivos são confirmados em outras pesquisas [60] que mostraram vantagens das redes neurais frente aos modelos autorregressivos, ficando claro que o modelo ARMA é mais um recurso comparativo do que um modelo de equivalência com as redes neurais artificiais.

Uma desvantagem do uso de RNAs para a modelagem reside na questão da abordagem tipo caixa preta. Nesses tipos de modelos, obtemos a relação entre uma ou diversas variáveis, não sendo possível, entretanto, identificar como essa relação foi obtida, ou seja, qual o processo físico que melhor descreve o fenômeno. Um outro ponto está na questão da disponibilidade dos dados ou no tamanho da série temporal. Se houver escassez de dados, isso pode prejudicar o aprendizado da RNA, tendo impacto nesse caso, na generalização da rede para eventos onde a rede não foi ensinada. Isso pode levar a um modelo fraco.

O uso dos algoritmos genéticos acoplado a uma RNA mostrou-se interessante permitindo um melhor desempenho da modelagem, entretanto, por outro lado tem um custo computacional adicional, exigindo uma etapa para iniciar a matriz de pesos. Isso deve ser considerado principalmente quando o tamanho do conjunto de dados para treinamento for muito grande, o que pode tornar o método muito caro computacionalmente.

Para trabalhos futuros recomenda-se a avaliação de SVM (support vector machine) para a modelagem do processo chuva-vazão em comparação com RNA. Também recomenda-se a combinação da RNA com os modelos ARMA para modelagem chuva-vazão, onde o modelo ARMA pode ser utilizado para prever a componente linear da série temporal, e a RNA utilizada para previsão da componente não-linear. Além disso é interessante a avaliação da metodologia proposta para outros cenários, incluindo bacias hidrográficas de diferentes áreas de drenagem.

REFERÊNCIAS

- [1] SOARES, A. da S. *Predição de séries temporais econômicas por meio de redes neurais artificiais e transformada wavelet: combinando modelo técnico e fundamentalista*. Dissertação (Mestrado) - Universidade Federal de São Carlos, São Carlos, 2008.
- [2] SAAE *Saneamento Ambiental: ciclo da água*. Disponível em: <http://www.saaeamparo.com.br/contas/dicas/ciclo_da_agua.html>. Acesso em: 28 Mar. 2010.
- [3] BACK, T.; HAMMEL, U.; SCHWEFEL, H. P. Evolutionary computation: comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, v. 1, p. 3–17, Abr. 1997.
- [4] BARP, A. R. B. *Modelagem chuva-vazão em bacias hidrográficas com suporte em redes neurais artificiais..* Tese (Doutorado)- Universidade Estadual de Campinas, Campinas, 1999.
- [5] BLACK ,P. E. *Watershed Hydrology*. 2. ed. Boca Raton: CRC Press, 1996.
- [6] CAMPOLO, M.; ANDREUSSI, P.; SOLDATI, A. River flood forecasting with a neural network model. *Water Resources Research*, v. 35, p.1191–1197, Abr. 1999.
- [7] CHATFIELD, C. *The Analysis of Time Series: an Introduction*. 5. ed. London: Chapman and Hall/CRC, 1996.
- [8] CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, v. 2, p.303–314, Dez. 1989.
- [9] DAVIE, T. *Fundamentals of Hydrology*. 2. ed.: Milton Park: Routledge, 2008.
- [10] DAWSON, C. W.; WILBY, R. An artificial neural network approach to rainfall runoff modelling. *Hydrological Science*, v. 43, p. 47–66, Ago. 1998.
- [11] AGÊNCIA NACIONAL DE ÁGUAS. *Glossário de termos hidrológicos*. Disponível em: <<http://www.ana.gov.br/portalsnirh/Downloads/tabid/77/Default.aspx>>. Acesso em 20. Dez. 2009.
- [12] SILVA , L. N. de C. *Análise e síntese de estratégias de aprendizado para redes neurais artificiais*. 1998. Dissertação (Mestrado) - Universidade Estadual de Campinas, Campinas, 1998.
- [13] OLIVEIRA, M. A. de. *Aplicação de redes neurais artificiais na análise de séries temporais econômico-financeiras*. 2007. Tese (Doutorado) - Universidade de São Paulo, São Paulo, 2007.
- [14] BRAGA, A. de P.; CARVALHO, A. P. de L. F. de; LUDEMIR, T. B. *Redes Neurais Artificiais*. 2. ed.. Rio de Janeiro: LTC, 2007.
- [15] INSTITUTO DE PESQUISAS HIDRÁULICAS UFRGS. *Modelo iphmen*. Disponível em: <<http://www.ufrgs.br/iph/>>. Acesso em: 15 Nov. 2010.

- [16] SANTOS, C. C. dos. *Modelagem de bacias urbanas com redes neurais artificiais*. 2001. Dissertação (Mestrado) - Universidade de São Paulo, São Paulo, 2001.
- [17] SAJIKUMAR, N.; THANDAVESWARA, B. S. A non-linear rainfall-runoff model using an artificial neural network. *Journal of Hidrology*, v. 216, p. 32–55, Out. 1999.
- [18] BEASLEY, D., BULL, D. R., MARTIN, R. R. An overview of genetic algorithms: part 2. *Research Topics*, v. 15, p. 170-181, 1993.
- [19] AMABIS, J. M., MARTHO, G. R. *Biologia dos organismos: classificação, estrutura e função nos seres vivos*. São Paulo: Moderna, 1997.
- [20] LALLAHEM, S.; MANIA, J. A nonlinear rainfall-runoff model using neural network technique: example in fractured porous media. *Mathematical and Computer Modelling*, v. 37, p.1047–1061, 2003.
- [21] RUMELHART, D. E.; McCLELLAND, J. L. *Parallel distributed processing*. Cambridge: The Mit Press, 1987. v.1.
- [22] MINNS, A. W.; HALL, M. J. Artificial neural networks as rainfall-runoff models. *Hydrological Sciences*, v. 41, p. 399–417, Jun. 1996.
- [23] GONÇALVES, A. R., BRUNETTO, M. A. de O. C. Um novo modelo híbrido baseado em otimização por colônia de formigas e redes neurais para identificação de indivíduos com dpoc. XI Congresso Nacional de Informática em Saúde, 2008.
- [24] PALMA NETO, L. G., NICOLETTI, M. do C. *Introdução as redes neurais construtivas*. São Carlos: EDUFSCAR, 2005.
- [25] ANDERS, U.; KORN, O.. Model selection in neural networks. *Neural Networks*, v. 12, p. 309–323, 1999.
- [26] TSOUKALAS, L. H.; UHRIG, R. E. *Fuzzy and neural approaches in engineering*. New York: John Wiley & Sons, 1997.
- [27] STOFFER, D. S.; SHUMWAY, R. H. *Time series analysis and its applications : with R examples*. New York: Springer, 2006.
- [28] BARREIRA, A. C.; ALVARENGA, R.; JARDIM, J. *Modelo híbrido baseado em sistemas inteligentes para detecção de intrusos em redes de computadores tcp-ip*. The International Conference of Forensic Computer Science. 2006. Disponível em: <<http://www.icofcs.org/2006/ICoFCS2006-pp07.pdf>>
- [29] ATIYA, A. F.; SHAHEEN, S. I. A comparison between neural-network forecasting techniques - case study: river flow forecasting. *IEEE Transactions on Neural Networks*, v. 10, p. 402–409, Mar. 1999.
- [30] NORVIG, P., RUSSELL, S. *Inteligência artificial*. 2. ed. Elsevier, 2005.
- [31] RAJURKAR, M. P.; KOTHYARI, U. C., CHAUBE, U.C. Modeling of the daily rainfall runoff relationship with artificial neural network. *Journal of Hidrology*, v. 285, p 96–113, Ago. 2003.

- [32] RIAD, S.; MANIA, J., BOUCHAOU, L.; NAJJAR, Y. Rainfall-runoff model using an artificial neural network approach. *Mathematical and Computer Modelling*, v. 40, p. 839–846, Out. 2004.
- [33] EHLERS, R. S. Análise de séries temporais. 2007. Disponível em: <http://www.each.usp.br/rvicente/AnaliseDeSeriesTemporais.pdf>.
- [34] ESHEL, G. The yule walker equations for the ar coefficients. Disponível em: <http://www.stat.wharton.upenn.edu/~steele/Courses/956/Resource/YWSourceFiles/YW-Eshel.pdf>>. Acesso em: 10. Out. 2009.
- [35] GOVINDARAJU, R. S. Artificial neural networks in hydrology. I: Preliminary concepts. *Journal of Hydrologic Engineering*, v. 5, p.115–123, Abr. 2000.
- [36] HALE, A. B. *Neuron, impulse, synapse*. Disponível em: http://www2.cedarcrest.edu/academic/bio/hale/bioT_EID/lectures/tetanus-neuron.html>. Acesso em: 17. Nov. 2009.
- [37] HAMZACEBI, C. Improving artificial neural networks performance in seasonal time series forecasting. *Information Sciences*, v. 178, p. 4550–4559, Jul. 2008.
- [38] HAYKIN, S. *Redes neurais: princípios e prática*. 2. ed. Porto Alegre: Bookman, 2008.
- [39] HEATON, J. *Encog artificial intelligence framework for java*. Disponível em <http://code.google.com/p/encog-java/>>. Acesso em 30. Jun. 2010.
- [40] HEATON, J. *Encog neural network tutorial*. Disponível em: <http://www.heatonresearch.com/taxonomy/term/42>>. Acesso em: 30. Jun. 2010.
- [41] HEATON, J. *Introduction to neural networks with Java*. 2. ed. Heaton Research Inc, 2008.
- [42] HO, S. A comparative study of neural network and box-jenkins arima modeling in time series prediction. *Computers & Industrial Engineering*, v. 42, p. 371–375, Abr. 2002.
- [43] IPARDES. *Hidrografia do estado do Paraná*. Disponível em: http://www.ipardes.gov.br/pdf/mapas/meio_ambiente/hidrografia.pdf>. Acesso em: 29. Mar. 2010.
- [44] KAPLAN, J. *Matlabcontrol: a java api to control and interact with matlab*. Disponível em: <http://code.google.com/p/matlabcontrol/>>. Acesso em 27. Jun. 2010.
- [45] KLOEPPPEL, B. D. *Bacia hidrográfica*. Disponível em: <http://paws.wcu.edu/bkloepfel/nrm460.html>>. Acesso em: 11. Nov. 2009.
- [46] SYLAIOS, G.; ZAFEIRIOU, E.; TSIHRINTZIS, V. A.; KOUTROUMANIDIS, T. Genetic modeling for the optimal forecasting of hydrologic time-series: application in nestos river. *Journal of Hydrology*, v. 368, p. 156–164, Abr. 2009.

- [47] LJUNG, L. *System identification toolbox: user guide*. The Mathworks Inc, 2010.
- [48] MACHADO, F. W. *Modelagem chuva-vazão mensal utilizando redes neurais artificiais*. 2005. Dissertação (Mestrado) - Universidades Federal do Paraná, Curitiba, 2005.
- [49] JAIN, A.; DANDY, G. C.; SUDHEER, K. P.; MAIER, H. R. Methods used for the development of neural networks for the prediction of water resource variables in river systems: current status and future directions. *Environmental Modelling & Software*, v. 25, p. 891–909, Ago. 2010.
- [50] THE MATHWORKS. *System identification toolbox 7.4*. Disponível em: <<http://www.mathworks.com/products/sysid/>>. Acesso em: 28. Jun. 2010.
- [51] NASSERI, M.; ASGHARI, K.; ABEDINI, M. J. Optimized scenario for rainfall forecasting using genetic algorithm coupled with artificial neural network. *Expert Systems with Applications*, v. 35, p. 1415–1421, Out. 2008.
- [52] REZENDE, S. O. *Sistemas inteligentes: fundamentos e aplicações*. Manole, 2005.
- [53] ROJAS, I.; HERRERA, L. J.; POMARES, H.; MARQUEZ, L.; PASADAS, M.; ROJAS, F.; VALENZUELA, O. Soft-computing techniques and arma model for time series prediction. *Neurocomputing*, v. 71, p. 519–537, Jan. 2008.
- [54] SEDKI, A.; OUAZAR, D.; EL MAZOUZI, E. Evolving neural network using real coded genetic algorithm for daily rainfall runoff forecasting. *Expert Systems with Applications*, v. 36, p. 4523–4527, Abr. 2009.
- [55] SHAMSELDIN, A. Y. Application of a neural network technique to rainfall-runoff modelling. *Journal of Hydrology*, v. 199, p. 272–294, 1997.
- [56] UNITED STATES GEOLOGICAL SURVEY. *Rio Baron fork at dutch mills*. Disponível em: <http://waterdata.usgs.gov/ok/nwis/uv?period=60&site_no=07196900>. Acesso em 03. Jan. 2010.
- [57] THIERENS, D. E.; DIRK; GOLDBERG. Convergence models of genetic algorithm selection schemes. In PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature, p. 119–129, 1994.
- [58] TOKAR, A. S.; JOHNSON, P. A. Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering*, v. 4, p. 232–239, 1999.
- [59] WEN-CHUAN, W. et al. A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. *Journal of Hydrology*, v. 374, p. 294–306, Ago. 2009.
- [60] WU, C. L.; CHAU, K. W., LI, Y. S. Methods to improve neural network performance in daily flows prediction. *Journal of Hydrology*, v. 372, p. 80–93, Jun. 2009.
- [61] XIN, Y. Evolving artificial neural networks. *Proceedings of the IEEE*, v. 87, p. 1423–1447, Set. 1999.