



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

JESSÉ DA COSTA ROCHA

**PREDIÇÃO DO CONSUMO DE ÁGUA POR
MEIO DE REDES NEURAIS ARTIFICIAIS –
UM ESTUDO DE CASO EM BELÉM-PA**

Belém – Pará

2018

JESSÉ DA COSTA ROCHA

**PREDIÇÃO DO CONSUMO DE ÁGUA POR
MEIO DE REDES NEURAIS ARTIFICIAIS –
UM ESTUDO DE CASO EM BELÉM-PA**

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação da Universidade Federal do Pará, como requisito para obtenção de título de Bacharel em Sistemas de Informação. Instituto de Ciências Exatas e Naturais.

Orientadora: Prof^a. Dr^a. Fabíola Pantoja Oliveira Araújo

Belém – Pará

2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

ROCHA, JESSÉ

PREDIÇÃO DO CONSUMO DE ÁGUA POR MEIO DE REDES NEURAIAS ARTIFICIAIS : UM ESTUDO DE CASO EM BELÉM-PA / JESSÉ ROCHA. - 2018.

51 f. : il. color.

Trabalho de Conclusão de Curso (Graduação) - Faculdade de Computação, Instituto de Ciências Exatas e Naturais, Universidade Federal do Pará, Belém, 2018.

Orientação: Profa. Dra. Fabíola Araújo

1. Redes Neurais Artificiais. 2. Inteligência Artificial. 3. Consumo de Água. 4. Variáveis Climatológicas. I. Araújo, Fabíola, *orient.* II. Título

CDD 006.32

JESSÉ DA COSTA ROCHA

PREDIÇÃO DO CONSUMO DE ÁGUA POR MEIO DE REDES NEURAS ARTIFICIAIS – UM ESTUDO DE CASO EM BELÉM-PA

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação da Universidade Federal do Pará, como requisito para obtenção de título de Bacharel em Sistemas de Informação. Instituto de Ciências Exatas e Naturais.

Trabalho aprovado. Belém – Pará, 02 de março de 2018.

Conceito:

BANCA EXAMINADORA

Prof^a. Dr^a. Fabíola Pantoja Oliveira Araújo
(Orientadora – UFPA)

Prof. Dr. Claudomiro de Souza de Sales Junior
(Membro Interno – UFPA)

Prof. Dr. Marcos César da Rocha Seruffo
(Membro Externo – UFPA)

Belém – Pará
2018

Agradecimentos

Agradeço aos meus pais Celene Rocha e José Rocha que sempre se esforçaram para me dar uma boa educação em todos os sentidos.

Aos meus irmãos Joselene Rocha e Josué Rocha pela amizade e incentivo. Especialmente ao Josué que foi o meu consultor de engenharia durante a elaboração deste trabalho.

A minha esposa Ádria Pacheco pelo amor e companheirismo e pela paciência que teve comigo durante o desenvolvimento deste trabalho.

Aos professores e funcionários da Faculdade de Computação da UFPA pelo conhecimento compartilhado. Especialmente para a minha orientadora professora Fabíola Araújo que sempre se mostrou pronta para me ajudar.

Aos meus colegas do curso de Sistemas de Informação das turmas de 2011, 2012 e 2013. Especialmente para o Isaac e o Leonardo que foram os amigos mais próximos nessa jornada.

Finalmente, a todas as pessoas que direta ou indiretamente contribuíram para que eu atingisse o objetivo de me formar no curso de Bacharelado em Sistemas de Informação.

Resumo

A escassez de água é um problema sério e constante em muitas cidades do Pará e do Brasil. Neste cenário, as várias técnicas de Inteligência Artificial, especialmente as Redes Neurais Artificiais, aparecem como alternativas capazes de auxiliar no planejamento e gerenciamento de Sistemas de Abastecimento de Água. Entre as muitas abordagens possíveis, optamos por analisar as variações no consumo de água em função do clima (temperatura, umidade e precipitação). Esta escolha é justificada porque o clima está entre os fatores que mais influenciam o consumo de água e também porque ainda há poucos estudos publicados nesta linha de pesquisa. Nesta perspectiva, o principal objetivo deste trabalho foi elaborar uma Rede de Neurais Artificiais para prever o consumo de água em um Sistema de Abastecimento localizado na cidade de Belém-PA e a principal motivação foi estimular a coleta e armazenamento de consumo de água, bem como o uso desses dados em sistemas inteligentes capazes de apoiar as decisões dos gestores de recursos hídricos na cidade de Belém e no Estado do Pará. Após quatro experimentos concluiu-se que a melhor arquitetura e configuração de RNA para resolver o problema proposto é uma rede com uma única camada escondida com 5 neurônios e função de ativação sigmoideal e uma camada de saída com um neurônio com função de ativação linear.

Palavras-chaves: Redes Neurais Artificiais, Inteligência Artificial, Consumo de Água e Variáveis Climatológicas.

Abstract

Water scarcity is a serious and constant problem in many cities in Pará and Brazil. In this scenario, the various techniques of Artificial Intelligence, especially Artificial Neural Networks, appear as alternatives capable of assisting in the planning and management of Water Supply Systems. Among the many possible approaches, we opted to analyze variations in water consumption as a function of climate (temperature, humidity and rainfall). This choice is justified because the climate is among the factors that most influence the consumption of water and also because there are still few studies published in this line of research. In this perspective, the main objective of this work was to elaborate an Artificial Neural Network to predict the water consumption in a Supply System located in the city of Belém-PA, and the main motivation was to stimulate the collection and storage of water consumption, as well as the use of this data in intelligent systems capable of supporting the decisions of the water resources managers in the city of Belém and in the State of Pará. After four experiments it was concluded that the best architecture and configuration of RNA to solve the proposed problem is a network with a single hidden layer with 5 neurons and sigmoidal activation function, and an output layer with a neuron with linear activation function.

Keywords: Artificial Neural Networks, Artificial Intelligence, Water Consumption and Climatological Variables.

Lista de ilustrações

Figura 1 – Unidades constituintes de um SAA.	15
Figura 2 – Crescimento Populacional do Brasil.	16
Figura 3 – Déficit de abastecimento de água.	18
Figura 4 – Localização do Setor Águas Negras.	20
Figura 5 – Neurônio Biológico.	26
Figura 6 – Neurônio Artificial.	27
Figura 7 – Fluxos das duas fases do treinamento.	31
Figura 8 – Parâmetros usados na criação da rede neural.	35
Figura 9 – Arquitetura da RNA criada.	36
Figura 10 – Parâmetros de treinamento.	37
Figura 11 – Resultado do treinamento da rede neural.	37
Figura 12 – Convergência do erro quadrático.	38
Figura 13 – Saídas finais obtidas.	39
Figura 14 – Histograma com os valores de saída obtidos.	39
Figura 15 – Resultado do teste.	40
Figura 16 – Histograma com os valores obtidos no teste.	40
Figura 17 – Análise do treinamento.	41
Figura 18 – Análise do teste.	42
Figura 19 – Arquitetura da 2ª RNA.	43
Figura 20 – Convergência do erro quadrático para o 2º experimento.	44
Figura 21 – Histograma do erro do 2º experimento.	45
Figura 22 – Convergência do erro quadrático para o 3º experimento.	46
Figura 23 – Convergência do erro quadrático para o 4º experimento.	47

Lista de tabelas

Tabela 1 – Resumo dos resultados dos experimentos.	47
--	----

Lista de abreviaturas e siglas

ABES	Associação Brasileira de Engenharia Sanitária e Ambiental
COSANPA	Companhia de Saneamento do Pará
ETAs	Estações de Tratamento de Água
Inmet	Instituto Nacional de Meteorologia
IA	Inteligência Artificial
Matlab	<i>Matrix Laboratory</i>
MSE	<i>Mean Squared Error</i>
MLP	<i>Multilayer Perceptron</i>
Nnstart	<i>Neural Network Start GUI</i>
Nntool	<i>Neural Network Toolbox graphical user interface</i>
ONU	Organização das Nações Unidas
OMS	Organização Mundial da Saúde
PDSAA	Plano Diretor do Sistema de Abastecimento de Água
RBF	<i>Radial Basis Functions</i>
RNA	Redes Neurais Artificiais
RMB	Região Metropolitana de Belém
RMSP	Região Metropolitana de São Paulo
RED	Reservatório Elevado de Distribuição
SI	Sistema Internacional de Unidades
SAA	Sistema de Abastecimento de Água
TANSIG	<i>Symmetric sigmoid transfer function</i>

Sumário

1	INTRODUÇÃO	12
1.1	Objetivo Geral	12
1.1.1	Objetivos específicos	13
1.2	Justificativa e Motivação	13
1.3	Organização do Trabalho	14
2	SISTEMAS DE ABASTECIMENTO DE ÁGUA	15
2.1	Conceitos e Partes Integrantes	15
2.2	Análise das Variações de Consumo	16
2.3	A Problemática do Abastecimento de Água na Região Metropolitana de Belém (RMB)	17
2.4	O Setor Águas Negras	20
2.4.1	Especificações Técnicas Do Setor Águas Negras	20
2.4.2	Descrição do Funcionamento do SAA do Setor Águas Negras	21
2.4.3	Dados Climatológicos Relacionados ao Setor Águas Negras	22
3	FUNDAMENTAÇÃO TEÓRICA DAS REDES NEURAIS	23
3.1	Aprendizado Supervisionado	23
3.2	Redes Neurais Artificiais	24
3.2.1	RNA: Uma Visão Geral	24
3.2.1.1	Breve Histórico Das RNAs	24
3.2.1.2	Neurônios Biológicos × Neurônios Artificiais	25
3.2.1.3	Funções de Ativação e Arquiteturas De RNAs	27
3.2.1.4	Aprendizado por Correção de Erro	28
3.2.2	Rede Perceptron de Múltiplas Camadas	28
3.2.2.1	Arquitetura de Redes MLP	29
3.2.2.2	Treinamento de Redes MLP	30
3.2.3	O Algoritmo <i>Back-propagation</i>	31
4	EXPERIMENTOS REALIZADOS	34
4.1	Primeiro Experimento	34
4.1.1	Pré-processamento dos Dados de Treino e Teste	34
4.1.2	Arquitetura e Parâmetros da Rede Neural Utilizada	35
4.1.3	Treino da RNA	36
4.1.4	Execução da RNA Com os Dados de Teste	40
4.1.5	Análise dos Resultados Obtidos no Treinamento e no Teste	41

4.2	Segundo Experimento	42
4.2.1	Pré-processamento dos Dados de Treino e Teste	42
4.2.2	Arquitetura e Parâmetros da Rede Neural Utilizada	42
4.2.3	Treino da RNA e Resultados	44
4.3	Terceiro Experimento	45
4.4	Quarto Experimento	46
5	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	48
	REFERÊNCIAS	50

1 Introdução

Nos bairros periféricos da Região Metropolitana de Belém (RMB) os moradores estão sendo afetados diariamente pela falta de água. Nesse cenário, dramaticamente marcado pelo desinvestimento e pela incompreensão da demanda reprimida, a Inteligência Artificial (IA), em especial as Redes Neurais Artificiais (RNA), pode ser utilizada no apoio à operação dos Sistemas de Abastecimento de Água (SAA), destacadamente na previsão de consumo diário.

Como se não bastassem as limitações intrínsecas aos SAAs, os erros na operação dos SAAs vem degradando ainda mais a qualidade de vida da população. Não obstante, a minimização das falhas pode ser conseguida com o uso de ferramentas de planejamento, tais como os modelos de previsão de curto prazo.

Esses modelos oferecem subsídio para a decisão sobre manobra operacional mais adequada. É nesse contexto que a determinação da variação do consumo de água ganha muita importância, já que os modelos de previsão de curto prazo baseiam-se na resposta instantânea dos sistemas a picos prolongados e flutuações acentuadas de demanda.

A abordagem não-tradicional utilizada pela IA é muito vantajosa, podendo ser aplicada nas companhias de saneamento que enfrentam problemas com a qualidade dos seus dados e dispõem de poucos recursos para a aquisição de equipamentos de medição, assim como para a qualificação da sua equipe técnica.

Nesse trabalho serão apresentados os procedimentos metodológicos e os resultados obtidos a partir da utilização de redes neurais artificiais para a análise da variação do consumo de água no Setor Águas Negras, Distrito de Icoaraci, na cidade de Belém-PA. A referida análise considerou como influenciadores do consumo de água os seguintes fatores climatológicos: temperatura, umidade e chuva.

Estudos realizados em outras regiões do Brasil demonstraram que o consumo de água é maior durante o verão e menor durante o inverno principalmente por causa da variação da temperatura, embora a relação entre consumo de água e temperatura não seja linear. De um modo geral a temperatura e a umidade tem apresentado as maiores correlações com o consumo. Não obstante, em função de comportamentos e de suas consequências indiretas, as chuvas também influenciam na variação do consumo de água.

1.1 Objetivo Geral

Este trabalho tem como principal objetivo utilizar Redes Neurais Artificiais (RNA) para prever o consumo de água no Setor Águas Negras, no Distrito de Icoaraci (Belém-PA)

em função de variáveis climatológicas (temperatura, umidade e chuva), e assim possibilitar uma melhor gestão do sistema de abastecimento de água do referido setor.

1.1.1 Objetivos específicos

- Levantar dados climatológicos (temperatura, umidade e chuva) e de consumo de água referentes ao Setor Águas Negras.
- Realizar o pré-processamento dos dados e gerar matrizes com as variáveis escolhidas para serem usadas na rede neural artificial.
- Elaborar redes neurais artificiais, com diferentes configurações, usando o Matlab para prever a demanda futura por água no referido setor.
- Analisar os resultados obtidos, identificando a arquitetura de rede neural mais adequada para o problema em questão.

1.2 Justificativa e Motivação

Os investimentos em automação dos Sistemas de Abastecimento de Água (SAA), visando principalmente à eficiência energética, cresceram consideravelmente nos últimos anos, mas esse instrumental (*hardware* e *software*) e os dados gerados por ele não estão sendo utilizados no planejamento das operações dos SAA.

O desequilíbrio entre a oferta limitada de água e o aumento acelerado da demanda reivindica uma operação cada vez mais eficiente das unidades de produção e distribuição de água. A interpretação das variações de consumo de água, assim como o entendimento dos seus principais influenciadores climatológicos, demográficos e socioeconômicos, são a chave para o desenvolvimento de modelos de previsão, capazes de suportar a operação requerida neste século.

Entretanto, ainda existem poucos estudos abordando essa temática. Numa análise feita de 1997 a 2004 sobre as publicações nos congressos internacionais da Associação Brasileira de Engenharia Sanitária e Ambiental (ABES), menos 1 % eram relacionados a análise de consumo de água. É bem verdade que a quantidade de trabalhos acadêmicos nessa área aumentou significativamente nos últimos anos, mesmo assim os artigos e pesquisas sobre estratégia e gestão no setor de saneamento ainda são diminutos.

Como acontece na maioria das cidades do Brasil, boa parte da população da Grande Belém sofre, em alguma medida, com a falta de água. Por conta disso, o presente trabalho busca contribuir com a solução desse problema crônico, usando recursos computacionais para prever futuras demandas por água e assim possibilitar um melhor planejamento estratégico nesse importante setor. Dessa forma, a principal motivação desse trabalho é

estimular a coleta e o armazenamento de dados relativos ao consumo de água, bem como o uso desses dados em sistemas inteligentes capazes de dar suporte às tomadas de decisão dos gestores dos SAA.

As principais contribuições deste trabalho são: melhoria na compreensão das relações existentes entre variáveis climatológicas e o consumo de água; criação de uma RNA capaz de auxiliar as tomadas de decisões dos gestores de SAA e estudo das ferramentas disponíveis no Matlab para a implementação de RNAs.

1.3 Organização do Trabalho

Este trabalho está dividido em cinco capítulos. O capítulo 1 é introdutório e os demais capítulos estão estruturados da seguinte maneira:

Capítulo 2: apresenta os conceitos básicos sobre consumo e distribuição de água. Em seguida, discorre sobre o consumo e a distribuição de água em Belém. Por fim, descreve as especificações técnicas e o funcionamento do Setor Águas Negras.

Capítulo 3: expõe a fundamentação teórica das RNAs em geral e em particular das RNAs multicamadas. Segue com uma explicação detalhada do algoritmo *back-propagation* e conclui com a apresentação de um trabalho correlato.

Capítulo 4: apresenta os experimentos realizados. Inicia com o pré-processamento dos dados, depois segue com a implementação das RNAs e finaliza como os resultados obtidos.

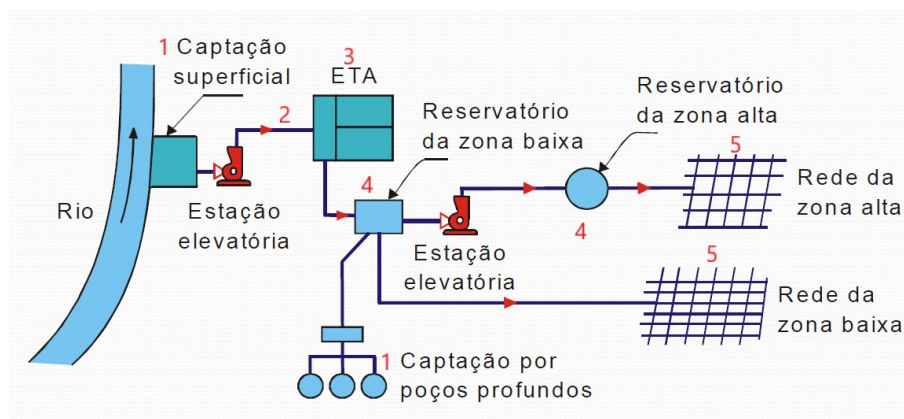
Capítulo 5: discute os resultados dos experimentos, faz as considerações finais e indica trabalhos futuros.

2 Sistemas de Abastecimento de Água

2.1 Conceitos e Partes Integrantes

Segundo Tsutiya (2006), o Sistema de Abastecimento de Água (SAA) é um conjunto de obras, equipamentos e serviços destinados ao abastecimento de água potável de uma determinada coletividade. As partes desse sistema estão numeradas na Figura 1. A explicação de cada uma dessas partes encontra-se logo abaixo da Figura 1.

Figura 1 – Unidades constituintes de um SAA.



Fonte – Adaptada de Tsutiya (2006).

1. Captação: a água bruta é captada em mananciais superficiais (rios e lagos) ou subterrâneos (poços tubulares);
2. Adução: a água captada dos mananciais é bombeada até as Estações de Tratamento de Água (ETAs);
3. Tratamento: por meio de uma série de processos físico-químicos, a água bruta é tornada potável;
4. Reservação: depois de tratada, a água é bombeada até reservatórios. A reservação é o armazenamento da água, entre o tratamento e o consumo, com os objetivos de: suprir as variações horárias de consumo, garantir a adequada pressurização do sistema de distribuição e garantir reservas de emergência para o caso de ocorrerem interrupções bruscas nos sistemas de bombeamento;
5. Distribuição: parte final do sistema, a água, através de uma rede de canalizações, é distribuída para o consumidor, pronta para uso;

Em um SAA, a água bruta é captada e em seguida direcionada para um tratamento específico, para que se torne potável. Depois de ter sido reservada, a água potável é distribuída a população por meio de uma rede de canalizações, em quantidade satisfatória para suprir sua necessidade de consumo.

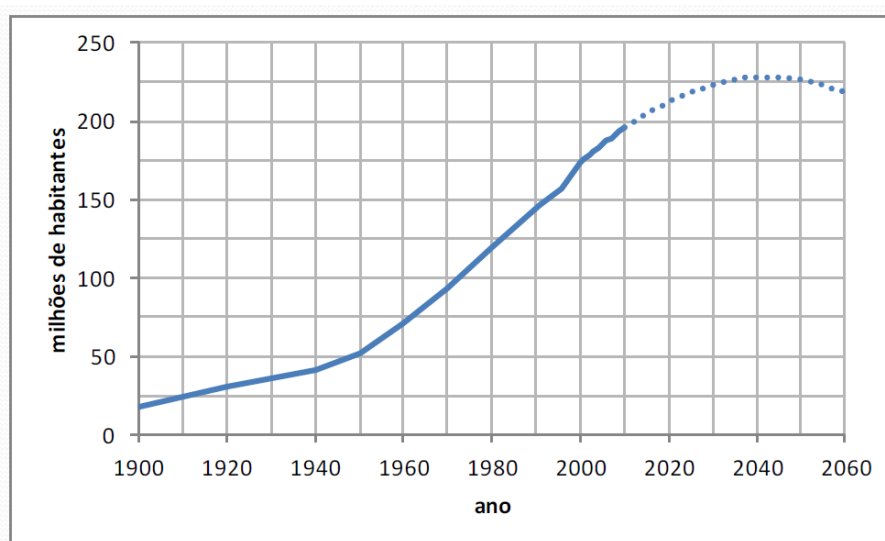
2.2 Análise das Variações de Consumo

A falta de água para o consumo humano deixou de ser uma ameaça e passou a ser uma realidade na maioria das grandes cidades do mundo. O crescimento vertiginoso da população mundial e a sua concentração em megalópoles tiveram como consequência a alteração da qualidade das águas dos mananciais de abastecimento público.

Dados do Fundo das Nações Unidas para a Infância (Unicef) e da Organização Mundial da Saúde (OMS) revelam que quase metade da população mundial (2,6 bilhões de pessoas) não conta com serviço de saneamento básico e que uma em cada seis pessoas (cerca de 1,1 bilhão de pessoas) ainda não possui sistema de abastecimento de água adequado.

Atualmente o Brasil possui uma população total de 204.450.649 habitantes (IBGE, 2015), a maior parte (84%) está concentrada em áreas urbanas demandando enormes quantidades de água para pequenas áreas desprovidas de mananciais com potencial para o abastecimento público. Uma realidade muito diferente daquela vivenciada nos anos de 1940, quando haviam apenas 03 moradores nas cidades para cada 10 brasileiros. Na Figura 2 é possível perceber o crescimento populacional acelerado que vem ocorrendo no Brasil desde a década de 50.

Figura 2 – Crescimento Populacional do Brasil.



Fonte – (IBGE, 2015).

Segundo a Organização das Nações Unidas (ONU), uma pessoa necessita de, pelo

menos, 40 litros de água por dia para satisfazer as suas necessidades básicas (beber, tomar banho, escovar os dentes, cozinhar etc). Não obstante, existe uma incrível variação no consumo diário de água ao redor do mundo. Estudos revelaram que um europeu consome, em média, 150 litros de água por dia, enquanto que um indiano apenas 25 litros. Além da disponibilidade local, o consumo médio de água está relacionado a diversos fatores, dentre eles:

- Condições climáticas;
- Nível de desenvolvimento da região;
- Renda familiar;
- Número de habitantes da residência;
- Características culturais da comunidade;
- Valor da tarifa de água;
- Estrutura e gestão do sistema de abastecimento de água;

No Brasil, onde a disponibilidade de água varia de uma região para outra, as reservas de água potável estão diminuindo. Dentre as principais causas da diminuição da oferta de água potável estão, principalmente, o aumento incontrolável do consumo, o desperdício e a poluição das águas superficiais e subterrâneas.

Segundo Iglésias, Gomes e García (2007), a solução para o problema da falta de água será mais efetiva se for orientada pela gestão da demanda, sem afetar a qualidade de vida das pessoas, e não pela gestão da oferta que depende da construção de novas estruturas (obras hidráulicas). Portanto, trata-se, antes de tudo, de ser mais eficiente na hora de consumir esse recurso tão valioso que é a água, e de diminuir as perdas nos próprios sistemas de distribuição de água.

2.3 A Problemática do Abastecimento de Água na Região Metropolitana de Belém (RMB)

Quando analisamos os números do saneamento no Brasil, relativos à distribuição regional do déficit de saneamento (Figura 3), a Região Norte aparece como a pior, bem atrás da Região Nordeste. O mais grave é que as pesquisas realizadas pelo IBGE apontam que, no período de 2000 a 2008, a ampliação da infraestrutura de saneamento na Região Norte foi inexpressiva, se comparada com a ocorrida nas outras Regiões do Brasil.

Figura 3 – Déficit de abastecimento de água.

REGIÃO	DÉFICIT - REDES DE ABASTECIMENTO DE ÁGUA (% de domicílios)	
	2000	2008
Norte	55,7	54,7
Nordeste	47,1	31,7
Sudeste	29,5	12,5
Sul	30,9	15,8
C. Oeste	33,7	18,0
Brasil	46,1	21,4

Fonte – (PNSB, 2008).

Estudos divulgados pelo Ministério das Cidades apontam que seriam necessários pelo menos R\$ 420 bilhões até o ano de 2030 para a universalização dos serviços de saneamento básico no Brasil, considerando abastecimento de água, esgotamento sanitário, manejo de águas pluviais e gerenciamento de resíduos sólidos (PLANSAB, 2013).

A Companhia de Saneamento do Pará (COSANPA), baseada no Plano Diretor do Sistema de Abastecimento de Água (PDSAA) da Região Metropolitana de Belém (RMB), captou cerca de R\$ 55 milhões de reais para a ampliação da capacidade de produção e tratamento de água superficial do Bolonha.

Não obstante, tal investimento refere-se apenas as primeiras intervenções propostas pelo PDSAA. Segundo estimativas apresentadas no V volume (atualização) do PDSAA será necessário um investimento de R\$ 780 milhões de reais, no período de 2007 a 2025, para proporcionar o incremento de 400 mil novas ligações de água na RMB.

A água produzida pela COSANPA atende cerca de 74,70 % (1.621.162 habitantes) da população urbana da RMB, segundo a própria COSANPA (2006). Aproximadamente 70 % dessa água é proveniente de mananciais superficiais, e o restante é retirado de poços tubulares (águas subterrâneas). Atualmente existem 40 setores de abastecimento de água, além de dezenas de microssistemas, alguns administrados diretamente por Prefeituras, como é o caso de Benevides e de Santa Bárbara do Pará. O crescimento desordenado da área urbana vem descaracterizando os limites de alguns desses setores.

Os 19 setores de abastecimento que utilizam água proveniente do mesmo manancial superficial (rio Guamá e os lagos Água Preta e Bolonha) são denominados de integrados, por terem unidades comuns de captação, adução, elevação e tratamento. Os outros setores por não apresentarem nenhuma unidade em comum são denominados de isolados.

Nos sistemas que utilizam estações de tratamento de água e de condicionamento, as unidades são compostas pelas seguintes etapas: desferrização, complexação, desinfecção e fluoretação. Por outro lado, a maioria dos sistemas isolados, alimentados por água subterrânea retirada de poços tubulares com capacidade variando entre 60 a 360 m³/h, são desprovidos de unidades de tratamento de Água.

Segundo o Plano Diretor de Abastecimento de Água COSANPA (2006), entre os principais problemas operacionais observados no SAA da RMB estão:

- Falta de isolamento dos setores integrados de abastecimento de água;
- Reduzida macromedição dos volumes (vazões) de água produzida, tratada e distribuída na RMB;
- Necessidade de instalação de hidrômetros nas ligações de água;
- Falta de registro da vazão da água encaminhada para áreas de invasão;
- Falta de compatibilidade entre as áreas dos cadastros técnico e comercial;
- Ocorrência de zonas de baixa pressão;
- Ocorrência de zonas de elevada pressão;
- Intermitência do fornecimento de água;
- Grande volume de água perdido e/ou desperdiçado;

A incompatibilidade entre os cadastros comercial e operacional e as interligações das redes de diferentes setores de abastecimento de água contribuem, significativamente, para o elevado índice de perdas no sistema de abastecimento de água da RMB, assim como para o surgimento de consumos não autorizados, ligações clandestinas, violação de hidrômetro, furto de água em hidrantes e desperdícios de água na operação.

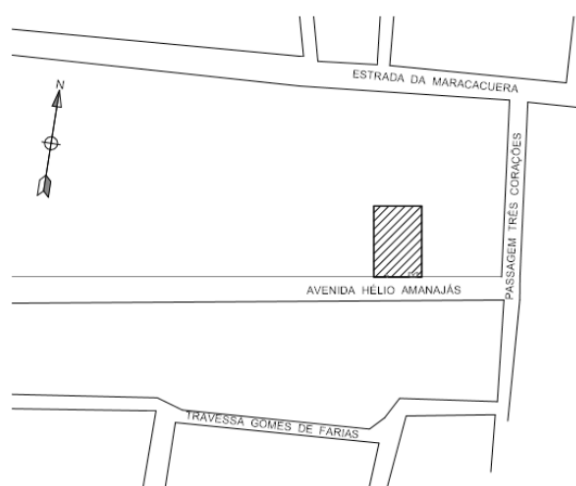
No Sistema Bolonha, Zona Central, o per capita da demanda de água chegou a 457 L/hab.d, sendo que destes 207 litros estavam relacionados a perda de água. Ainda segundo o Plano Diretor COSANPA (2006), no sistema de abastecimento de água da RMB os volumes perdidos de água se devem principalmente a: desperdício de água (especialmente em locais sem hidrometração), vazamentos em redes e adutoras de água e usos irregulares de equipamentos de abastecimento.

2.4 O Setor Águas Negras

O Setor Águas Negras está localizado no Distrito de Icoaraci, município de Belém, no Estado de Pará. O setor abrange os bairros de Recanto Verde, Águas Negras, Ocupação do Riso e 7 de Setembro, que juntos perfazem uma população total de 14.650 habitantes.

O SAA Setor Águas Negras está situado na área do Parque Verde (Figura 4). O referido Setor é operado pela Companhia de Saneamento do Pará (COSANPA) e possui área total equivalente a 651 m².

Figura 4 – Localização do Setor Águas Negras.



Fonte – Elaborada pelo autor.

Atualmente o SAA atende 14.650 pessoas através de 3.041 ligações domiciliares. O sistema funciona 24 horas, sendo que o conjunto motobomba funciona em média 18 horas/dia, mantendo o nível de água no reservatório favorável ao abastecimento. A vazão distribuída é da ordem de 234 m³/h.

Em Fevereiro de 2017 a operação do SAA foi automatizada e em junho do mesmo ano foi implantado o sistema Telemetrix para disponibilização de dados *on-line* sobre o funcionamento do SAA. Os dados sobre vazão de bombeamento e nível de água no reservatório são obtidos de minuto em minuto, totalizando 1.440 dados por dia de cada uma dessas variáveis diretamente relacionadas ao consumo de água.

2.4.1 Especificações Técnicas Do Setor Águas Negras

O SAA possui um poço tubular de 270,00 metros de profundidade revestido com tubos de aço-carbono de diâmetros 12” e 8”. O filtro possui comprimento de 39,24 metros de aço inox AISI-304 com diâmetro de 8”.

O sistema de bombeamento é constituído por uma bomba submersa LEÃO, Modelo: S270R-03, Motor: 01/100/2238TR, Potência: 100 CV, 380 V, Vazão: 240 m³/h e Altura Manométrica de 83,00 m.c.a. O acionamento do conjunto motobomba é realizado pelo operador da estação por meio de um painel de comando elétrico *soft start* – ssw 03. O conjunto é alimentado por um transformador 150 KVA–380 V.

O sistema conta com um Reservatório Elevado de Distribuição (RED), com capacidade de 500 m³. A altura total da torre é de 26,70 m. A rede de distribuição possui uma extensão total superior a 25 km (25.355 m) em PVC PBA, com diâmetros variando de 50 mm a 200 mm. No sistema não existe ETA. Sendo assim, atualmente não está sendo feito nenhum tipo de tratamento com produtos químicos.

2.4.2 Descrição do Funcionamento do SAA do Setor Águas Negras

De um modo geral, num SAA o volume do Reservatório Elevado de Distribuição (RED) varia acentuadamente em função do coeficiente da hora de maior consumo (GOMES, 2009). Nessas circunstâncias, o volume útil do RED é definido como sendo aquele compreendido entre o limite baixo e o limite alto.

Para análise da operação de um reservatório é utilizada, quando existente, a curva de demanda do dia de maior consumo. No caso do SAA – Setor Águas Negras – foi desenvolvida uma programação que aciona o conjunto motobomba sempre que o nível da água dentro do (RED) chega em 3 metros. Do mesmo modo, o sistema de bombeamento é desligado quando o nível da água no RED atinge 8 metros.

Foi assim que, medindo a variação dinâmica do nível de água no RED, o consumo de água foi determinado para cada período de 24 horas de funcionamento do Sistema. As medições foram realizadas automaticamente a cada minuto. Seguem abaixo alguns exemplos de variações ocorridas no dia 01 de Outubro de 2017.

EX₁ às 00h01 min

Operação: conjunto motobomba desligado (Q=0 m³/h), nível de água no RED 5,44 m.

Rebaixamento em 01 minuto: 0,03 m.

Consumo: 1,50 m³/min.

EX₂ às 02h17 min

Operação: conjunto motobomba ligado (Q=175 m³/h), nível de água no RED 3,02 m.

Rebaixamento em 01 minuto: 0,02 m.

Consumo: 3,42 m³/min.

EX₂ às 02h18 min

Operação: conjunto motobomba ligado (Q=190 m³/h), nível de água no RED 3,03 m.

Elevação em 01 minuto: 0,01 m.

Consumo: 3,67 m³/min.

2.4.3 Dados Climatológicos Relacionados ao Setor Águas Negras

O clima no Setor Águas Negras é quente e úmido com precipitação média anual alcançando os 2.834 mm. A temperatura média varia pouco durante o ano, atingindo 25° C em fevereiro e 26° C em novembro. Está na zona climática Af_i (classificação de Köppen), que coincide com o clima de floresta tropical, permanentemente úmida, com ausência de estação fria e temperatura do mês menos quente acima de 18° C.

Os dados climatológicos usados no presente trabalho foram obtidos no site oficial do Instituto Nacional de Meteorologia (Inmet) que é um instituto centenário ligado diretamente ao Ministério da Agricultura, Pecuária e Abastecimento (INMET, 2017). O referido instituto possui uma estação meteorológica no município de Belém, onde coleta dados climatológicos, tais como: temperatura, umidade, chuva, pressão, vento e radiação solar. Esses dados são disponibilizados gratuitamente no site do Inmet a todos os interessados. Dentre os dados disponíveis, foram selecionadas as variáveis temperatura, umidade e chuva, por serem as mais relevantes para o presente estudo. As referidas informações estão tabeladas de hora em hora. A temperatura do ar está registrada em grau Celsius, que é a escala usada na maioria dos países, embora a escala adotada pelo Sistema Internacional de Unidades (SI) seja a Kelvin. A umidade relativa do ar está anotada em % e representa a relação entre a quantidade de vapor de água presente no ar e a quantidade máxima de vapor de água que o ar pode absorver em uma dada temperatura. Logo 100% de umidade relativa do ar é o ponto de saturação a partir do qual o ar não pode mais absorver vapor de água. É importante observar que em ambientes com umidade alta, como é o caso de Belém, a sensação térmica é maior, isso porque a umidade alta dificulta a evaporação do suor que é o principal mecanismo de regulação da temperatura do corpo humano. A quantidade de chuva, por sua vez, está registrada mm. Uma pluviosidade de 1 mm equivale ao volume de 1 L de água de chuva que se acumulou em uma superfície de área igual a 1 m².

3 Fundamentação Teórica das Redes Neurais

3.1 Aprendizado Supervisionado

Aprender com exemplos significa melhorar o desempenho em uma determinada tarefa após receber informações específicas sobre o universo no qual a ação está sendo desenvolvida. A definição anterior é ampla porque existem vários tipos de aprendizagem por meio de exemplo. Contudo, nesta seção será abordado apenas um tipo de aprendizado, isto é, aprender a prever eventos futuros com base em uma série histórica. Em outras palavras: “a partir de uma coleção de pares de entrada e saída, aprender uma função que prevê a saída para novas entradas” (RUSSELL; NORVIG, 2013, p. 827).

A melhoria na performance, por sua vez, depende fundamentalmente de dois fatores: 1) A estrutura dos dados que estão a disposição do agente. 2) O tipo de *feedback* que está disponível para o agente. Os dados de entrada de um sistema inteligente podem ter diferentes estruturas, tais como: sentenças lógicas, vetores numéricos, vetores contendo valores qualitativos etc. Quanto ao *feedback* disponível, o processo de aprendizagem pode ser classificado como não supervisionado, por reforço ou supervisionado. No aprendizado não supervisionado o agente recebe padrões na entrada e os agrupa por categorias sem receber nenhum *feedback* explícito. No aprendizado por reforço, o agente recebe reforços positivos (recompensas) para as ações consideradas corretas e reforços negativos (punições) para as ações consideradas incorretas. No aprendizado supervisionado, por sua vez, o agente recebe exemplos de valores de entradas com suas saídas correspondentes e procura encontrar uma função que mapeie as entradas e suas respectivas saídas.

Segundo Russell e Norvig (2013), a aprendizagem supervisionada pode ser dividida em dois momentos principais: treinamento e teste. No treinamento é repassado um conjunto de pares ordenados (x, y) , sendo y uma função de x , ou seja, $y = f(x)$. Vale lembrar que y também pode ser uma função de múltiplas variáveis. O treinamento prossegue até que uma boa aproximação da função real $f(x)$ seja encontrada. A função encontrada é chamada de h , por ser uma hipótese, ou seja, uma possível solução para o problema, mas, como toda hipótese, a função h precisa ser testada antes de ser aceita. Durante o teste é repassado um novo conjunto de pares ordenados, distinto do primeiro. Nesse momento é verificado se para cada par ordenado do conjunto de teste é válida a relação $h(x) \simeq y$. Se a relação for válida, h passa a ser considerada uma boa generalização. Em todo caso, o que se busca é uma aproximação razoável para y , dado que encontrar o valor exato é praticamente impossível.

Quando o vetor y é apenas numérico, por exemplo $y = \{3; 5, 4; 7\}$, o problema

de aprendizagem é chamado de regressão. Por outro lado, quando o vetor y é do tipo *string* ou booleano, por exemplo $y = \text{baixo, médio, alto}$, a aprendizagem é chamada de classificação. Em algumas situações, existem várias hipóteses h consistentes com os dados fornecidos. Nessas situações, a hipótese mais simples deve ser escolhida. Tal escolha pode ser justificada por duas razões. Em primeiro lugar, hipóteses mais simples frequentemente são generalizações melhores e, dessa forma, deverão responder melhor às experiências futuras. Além disso, aplicar funções mais simples é sempre mais fácil.

3.2 Redes Neurais Artificiais

Uma das principais técnicas usadas atualmente para fazer um computador aprender por meio de exemplos é a rede neural artificial (RNA). Uma RNA pode aprender utilizando um dos três diferentes métodos de aprendizagem descritos anteriormente, isto é, o aprendizado por meio de reforço, o não supervisionado e o supervisionado. Neste ponto, vale ressaltar que todas as RNAs usadas no presente estudo foram treinadas por meio de um aprendizado supervisionado.

Os dois tópicos seguintes são dedicados à análise das RNAs. Primeiramente é apresentada uma visão geral das RNAs contendo a origem histórica e os conceitos fundamentais. Em seguida é exposto um tipo específico de RNA chamado de rede Perceptron de múltiplas camadas.

3.2.1 RNA: Uma Visão Geral

Este tópico apresenta uma breve história das RNAs e explica os seus conceitos básicos.

3.2.1.1 Breve Histórico Das RNAs

De acordo com Braga, Carvalho e Ludemir (2016), a história das RNAs começou em 1943 quando Warren McCulloch e Walter Pitts criaram o primeiro modelo de neurônio artificial de que se tem notícia. McCulloch era psicólogo e neurofisiologista. Pitts, por sua vez, era matemático. Os dois pesquisadores apresentaram à comunidade científica um modelo lógico-matemático de um neurônio artificial que foi chamado de neurônio MCP (iniciais dos sobrenomes dos inventores), junto com a descrição de sua capacidade computacional. O principal objetivo desses pesquisadores era criar um modelo para explicar o funcionamento do sistema nervoso.

Em 1949, Donald Hebb apresentou um estudo detalhado sobre como um neurônio biológico aprende e propôs aplicar o mesmo processo ao neurônio artificial. Segundo ele, um neurônio aprende por meio da variação dos pesos de suas entradas, ou seja, reforçando ou inibindo as conexões sinápticas com outros neurônios.

Em 1958, Frank Rosenblatt apresentou um novo modelo de neurônio artificial, o Perceptron. O Perceptron era basicamente um neurônio MCP com pesos de entrada ajustáveis. Por meio desse artifício era possível treinar o Perceptron para funcionar como um separador linear, isto é, separar, por meio de uma linha reta, os valores de entrada em duas classes distintas.

Contudo, em 1969, Minsky e Papert publicaram um estudo demonstrando que o Perceptron serviria apenas e tão somente como um separador linear, sendo incapaz de aprender funções mais complexas. Esse estudo desencorajou novas pesquisas em RNA durante toda a década de 70 do século passado.

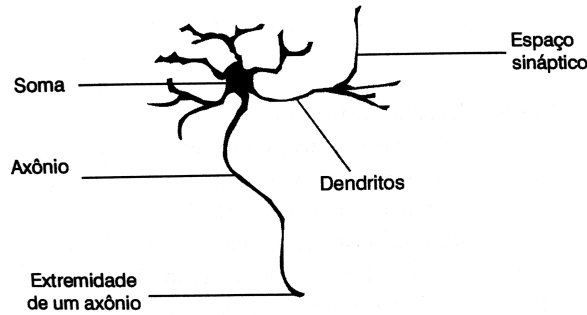
Foi só em 1982 que as pesquisas em RNA foram retomadas, após John Hopfield publicar um artigo mostrando as propriedades das associações de neurônios artificiais. Isso abriu caminho para o surgimento das redes *Perceptron* de múltiplas camadas, capazes de aprender funções não-lineares complexas. O desenvolvimento posterior do algoritmo de treinamento *back-propagation*, eficiente para as redes de múltiplas camadas, lançou as bases para todos os avanços que se seguiram.

3.2.1.2 Neurônios Biológicos × Neurônios Artificiais

Conforme abordado anteriormente, o desenvolvimento dos neurônios artificiais foi baseado nos neurônios biológicos. Entretanto, cabe ainda destacar quais são as semelhanças e as diferenças existentes entre esses dois modelos.

O neurônio biológico possui três partes principais: o corpo celular (soma), os dendritos e o axônio, como mostra a Figura 5. Os dendritos são as entradas do neurônio que recebem os impulsos nervosos vindos de outros neurônios. Dos dendritos os impulsos passam até chegarem no corpo celular. No corpo celular os impulsos são processados e, de acordo como o caso, enviados ou não para outros neurônios. O axônio, por sua vez, é o terminal de saída do neurônio. É por meio dele que o impulso nervoso é enviado para outros neurônios. A conexão entre o axônio de um neurônio anterior e o dendrito de um neurônio posterior é feita por meio de uma sinapse. Na sinapse, neurotransmissores são liberados pelo axônio anterior e recebido pelo dendrito do neurônio posterior. Certos tipos de neurotransmissores tem a função de gerar um impulso nervoso, enquanto que outros tipos de neurotransmissores tem a função oposta, ou seja, inibir a propagação de um impulso nervoso. Sendo assim, “As sinapses funcionam como válvulas, e são capazes de controlar a transmissão de impulsos – isto é, o fluxo da informação – entre os neurônios na rede neural” (BRAGA; CARVALHO; LUDEMIR, 2016, p. 6).

Figura 5 – Neurônio Biológico.



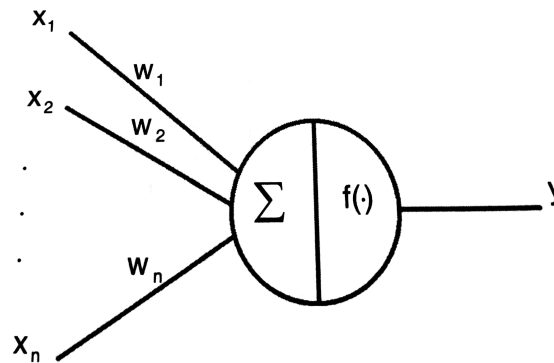
Fonte – (BRAGA; CARVALHO; LUDEMIR, 2016).

É importante ressaltar que o corpo do neurônio combina os diversos impulsos nervosos recebidos pelos dendritos e, caso o resultado ultrapasse um certo limiar de excitação, transmite um impulso por meio do axônio. Em outras palavras, quanto maior o acúmulo de fortes impulsos nas entradas do neurônio, maior é a probabilidade de o neurônio disparar um impulso para o neurônio seguinte.

Por outro lado, o neurônio artificial concebido por McCulloch e Pitts, ou seja, o modelo MCP reproduz de forma simplificada a estrutura e o funcionamento do neurônio biológico, conforme mostrado na Figura 6. Nele os dendritos são substituídos por terminais de entrada ($X_1, X_2, X_3, \dots, X_n$). O axônio é substituído por um terminal de saída (Y). As sinapses são substituídas por pesos associados aos terminais de entrada ($W_1, W_2, W_3, \dots, W_n$). Os pesos com valores positivos representam as sinapses excitatórias e os pesos com valores negativos representam as sinapses inibitórias. A combinação dos impulsos nervosos recebidos pelos dendritos é substituída pela soma ponderada das entradas, como mostra a Equação (3.1). Por fim, o limiar de excitação é substituído por uma função de ativação. O funcionamento da referida função será objeto de estudo do próximo tópico junto com as principais arquiteturas de RNAs. Antes porém, é preciso destacar uma importante diferença entre as redes neurais biológicas e as redes neurais artificiais. Nas redes neurais artificiais, todos os neurônios de uma mesma camada são ativados simultaneamente. Nas redes neurais biológicas, ao contrário, os neurônios de uma mesma camada podem ser acionados em tempos distintos uma vez que não há um clock central para sincronizar todos os neurônios.

$$\sum_{i=1}^n X_i W_i \quad (3.1)$$

Figura 6 – Neurônio Artificial.



Fonte – (BRAGA; CARVALHO; LUDEMIR, 2016).

3.2.1.3 Funções de Ativação e Arquiteturas De RNAs

Dentre os diferentes tipos de funções de ativação, quatro merecem destaque: degrau, sigmoidal, linear e gaussiana. A função de ativação usada pelo neurônio MCP é a degrau. Ela funciona como mostrado na Equação (3.2), na qual q representa o limiar de ativação escolhido.

$$f(x) = 1 \text{ se } \sum_{i=1}^n X_i W_i \geq q, \text{ ou } f(x) = 0 \text{ se } \sum_{i=1}^n X_i W_i < q \quad (3.2)$$

A função de ativação sigmoidal é uma aproximação contínua da função degrau, ou seja, permite uma transição suave entre o mínimo=0 e o máximo=1, sem sofrer nenhuma descontinuidade. Nisso ela difere da função degrau que é descontínua e apenas serve para ligar ou desligar o neurônio. A Equação (3.3) descreve a função sigmoidal. Essa função de ativação é a mais usada em RNAs de múltiplas camadas porque facilita a aproximação de funções contínuas.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.3)$$

Uma função de ativação linear do tipo $f(x) = x$ também pode ser usada para resolver problemas simples, mas nesse caso a RNA fica reduzida a uma espécie de calculadora de álgebra linear que multiplica o vetor de entrada pelo vetor dos respectivos pesos e depois soma o resultado.

Por fim, a função gaussiana é usada nas RNAs do tipo RBF (*Radial Basis Functions*) que estão fora do escopo deste trabalho. Em todo caso, a Equação (3.4) descreve essa função, em que m é o ponto médio e r é o raio de abertura da função.

$$f(x) = e^{-\frac{(x-m)^2}{r^2}} \quad (3.4)$$

As arquiteturas de RNAs, por sua vez, podem ser divididas em dois grandes grupos: redes *feedforward* e redes com recorrência. As redes *feedforward* podem ter uma ou múltiplas

camadas. Elas possuem esse nome porque os sinais de entrada sempre fluem da esquerda para a direita, ou seja, elas sempre são alimentadas para a frente (*feedforward*). Uma rede *feedforward* de uma única camada já é capaz de resolver problemas com múltiplas variáveis, mas com restrições quanto à complexidade. Por outro lado, basta acrescentar uma camada intermediária para que uma rede *feedforward* seja capaz de aproximar qualquer função contínua, independente da sua complexidade. Essas redes são consideradas estáticas porque as saídas dependem apenas dos valores atuais das entradas. Neste trabalho foram usadas apenas redes *feedforwards*. Nas redes recorrentes, ao contrário, os valores de saída dependem dos valores atuais de entrada e também dos valores atuais da saída uma vez que a saída é reconectada com as entradas de neurônios de camadas anteriores.

3.2.1.4 Aprendizado por Correção de Erro

As RNAs que aprendem de forma supervisionada utilizam um algoritmo de treinamento que busca minimizar o erro, ou seja, a diferença entre a saída esperada e a saída calculada pela rede. Em termos matemáticos, o erro pode ser definido como na Equação (3.5), na qual $y_d(t)$ é a saída desejada e $y(t)$ é a saída atual calculada pela rede.

$$e(t) = y_d(t) - y(t) \quad (3.5)$$

Para minimizar o erro é necessário ajustar os pesos da rede de forma incremental, isto é, a cada interação, os pesos sofrem pequenos ajustes. Esses ajustes são calculados em função do erro obtido $e(t)$, da taxa de aprendizado α , que determina o tamanho do passo que será dado em direção ao erro mínimo, e da entrada de cada neurônio. A Equação (3.6) é uma forma genérica para atualizar os pesos de uma RNA e é utilizada, com algumas modificações, na maioria dos algoritmos de aprendizado por correção de erro. Nela $w_i(t)$ e $x_i(t)$ correspondem respectivamente ao peso e a entrada i do neurônio. É importante notar que o ajuste do peso deve ser proporcional ao produto do erro pelo valor da entrada. Contudo, como será mostrado no tópico seguinte, RNAs de múltiplas camadas possuem uma equação de atualização dos pesos mais complexa que utiliza o gradiente de erro no lugar do erro $e(t)$.

$$w_i(t + 1) = w_i(t) + \alpha e(t) x_i(t) \quad (3.6)$$

3.2.2 Rede Perceptron de Múltiplas Camadas

Conforme mencionado na breve história das RNAs exposta anteriormente, a rede Perceptron de múltiplas camadas, em inglês MLP (*Multilayer Perceptron*), foi desenvolvida para superar a limitação do Perceptron de camada única. O Perceptron simples possui apenas uma camada e uma função de ativação degrau. Isso limita o Perceptron simples a ser um mero separador linear. A explicação é a seguinte: a saída do Perceptron é 1 se $\sum_{i=1}^n X_i W_i \geq q$, ou 0 se $\sum_{i=1}^n X_i W_i < q$, conforme mostrado na Equação (3.2).

Considerando, por exemplo, um Perceptron com duas entradas (x_1 e x_2) chega-se a equação da reta $x_1w_1 + x_2w_2 - q = 0$ que divide o espaço de entrada em duas partes. A primeira é a classe A_1 que está acima da reta e a segunda é a classe A_2 que está abaixo da reta.

Para superar a limitação do Perceptron simples é preciso introduzir a não-linearidade na RNA de duas maneiras: por meio de uma função de ativação não-linear e por meio do uso de múltiplas camadas. A RNA do tipo MLP satisfaz essas duas exigências, o que a torna potencialmente capaz de modelar qualquer fenômeno do mundo real, que são na maioria não-lineares, desde que se encontre uma estrutura de dados adequada para representá-lo. Uma MLP com função de ativação sigmoideal e uma camada intermediária é capaz de aproximar qualquer função contínua, linear ou não. Por sua vez, uma MLP com função de ativação sigmoideal e duas camadas intermediárias é capaz de aproximar qualquer função, contínua ou não. Isso acontece, como será detalhado a seguir, porque as camadas intermediárias de uma MLP transformam “o problema descrito pelo conjunto de dados no espaço de entrada em uma representação tratável para a camada de saída da rede” (BRAGA; CARVALHO; LUDEMIR, 2016, p. 67).

3.2.2.1 Arquitetura de Redes MLP

Uma rede MLP típica com uma camada de entrada, uma camada intermediária e uma camada de saída realiza duas transformações sucessivas para resolver um problema de classificação não-linear. A primeira transformação é feita na camada intermediária. Nela uma função do tipo $H(x; w_H)$, em que x representa o vetor com os valores de entrada e w_H o vetor com os pesos da camada intermediária, transforma a superfície não-linear apresentada na entrada da rede em uma superfície linear. Na camada de saída, uma função $Y(H(x; w_H); w_s)$, em que w_s representa o vetor com os pesos da camada de saída e $H(x; w_H)$ a saída da camada anterior, executa a separação linear. É importante destacar que a maioria dos problemas encontrados na realidade pode ser resolvido por uma MLP com apenas uma camada intermediária e que o uso de camadas intermediárias desnecessárias pode dificultar a convergência da rede para um erro mínimo.

Um outro importante parâmetro a ser considerado em uma rede MLP é a função de ativação. Em uma rede MLP as funções de ativação para as camadas intermediárias devem ser não-lineares. Dentre todas as funções de ativação não-lineares possíveis, a sigmoideal é preferida porque possui uma derivada simples de ser calculada. Essa derivada, por sua vez, é usada no cálculo do gradiente de erro, como será mostrado posteriormente. Por outro lado, a função de ativação da camada de saída pode ser linear ou não-linear, não havendo portanto nenhuma restrição nesse caso.

O parâmetro mais complexo a ser definido em uma rede MLP é sem dúvida a quantidade de neurônios na camada intermediária. Apesar das inúmeras pesquisas

realizadas nessa área, não existe uma resposta definitiva para essa questão. O que se sabe é que o número de neurônios na camada intermediária determina a complexidade da rede e, por conseguinte, a complexidade dos problemas que a rede é capaz de resolver. Sendo assim, a quantidade de neurônios na camada intermediária deve ser proporcional à complexidade do problema que se quer resolver. Esse princípio geral, contudo, não esgota a questão uma vez que na maioria dos casos a complexidade do problema a ser resolvido não é conhecida plenamente até que ele seja resolvido. Para contornar essa dificuldade, foram desenvolvidas diferentes abordagens, dentre as quais vale destacar a abordagem multiobjetivo. Segundo essa abordagem, o projeto de uma RNA deve ser orientado por dois objetivos básicos: em primeiro lugar, a RNA projetada deve ser capaz de convergir para um erro mínimo; em segundo lugar, a RNA deve possuir a menor complexidade possível capaz de atingir o primeiro objetivo. Para atingir esses dois objetivos simultaneamente, o projetista da RNA deve usar o método das tentativas, isso é, projetar uma RNA com uma complexidade mínima e verificar se ela converge para o erro mínimo. Caso isso não aconteça, o projetista deve aumentar gradativamente a complexidade da rede até que ela convirja para o erro mínimo.

A implementação de uma RNA com complexidade acima do necessário, isto é, o superdimensionamento de uma RNA traz consigo dois problemas. O primeiro e mais evidente é o desnecessário aumento do custo computacional. O segundo e mais grave é o indesejado aumento do universo das possíveis soluções que atendem ao critério de minimização do erro, sem que a quantidade de possíveis soluções que se aproximam da função geradora dos dados aumente. Desta forma, quanto maior o número de neurônios desnecessários, maior é a dificuldade para se achar uma solução ótima. Pode-se dizer que o superdimensionamento de uma RNA cria uma espécie de paradoxo da escolha, no qual o aumento das possibilidades de escolha aumenta não apenas a dificuldade da escolha, como também as chances de se fazer uma escolha ruim. Isso reforça a ideia de que funções mais simples costumam ser generalizações melhores, enquanto que funções mais complexas se ajustam melhor aos dados de teste, mas se afastam de uma melhor generalização ao tentarem colocar todos os pontos dentro da curva, inclusive os ruídos presentes na base de dados.

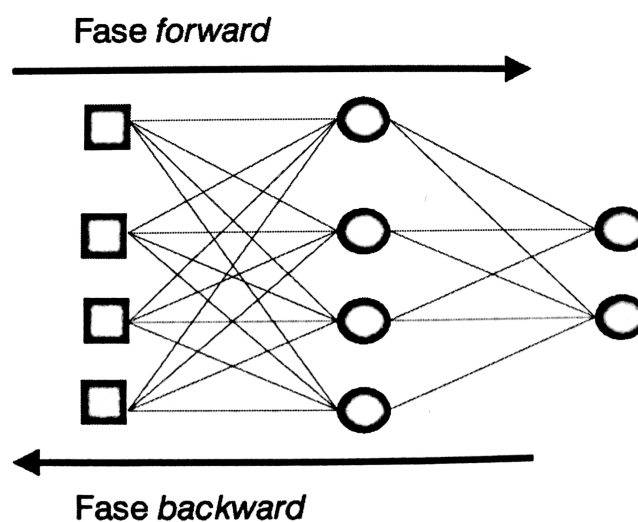
3.2.2.2 Treinamento de Redes MLP

Em linhas gerais, o treinamento de redes MLP é dividido em dois momentos distintos. O primeiro momento é chamado de fase *forward* e o segundo momento é chamado de fase *backward*. Na fase *forward*, os dados de entrada são propagados na rede da esquerda para a direita, passando de camada em camada até que a saída final seja produzida. Na fase *backward*, o erro é propagado da direita para a esquerda e com base nesse erro os pesos da rede são ajustados. Esse processo se repete até que um erro mínimo seja atingido.

Para mais detalhes, a fase *forward* pode ser dividida em três passos. 1º) As entradas da rede recebem o vetor de entrada e as saídas dos neurônios da primeira camada intermediária são calculadas. 2º) As saídas dos neurônios da primeira camada intermediária passam a ser as entradas da camada seguinte, e as saídas dos neurônios dessa camada são calculadas. O processo é repetido até chegar na camada de saída. 3º) A saída do neurônio da camada de saída é confrontada com o valor esperado e o erro é calculado.

A fase *backward*, por sua vez, pode ser dividida em quatro passos. 1º) O erro encontrado na fase anterior é usado para calcular o gradiente de erro. Em seguida, o gradiente de erro é utilizado para ajustar diretamente os pesos do neurônio da camada de saída. 2º) O gradiente de erro da camada de saída é propagado para os neurônios da camada intermediária. Para que isso ocorra com propriedade, o gradiente de erro é multiplicado pelo peso existente entre o neurônio da camada de saída e o neurônio da camada intermediária. O processo se repete para cada neurônio da camada intermediária. 3º) O erro propagado no passo anterior é usado para calcular o gradiente de erro de cada neurônio da camada intermediária. Posteriormente, os pesos de cada neurônio da camada intermediária são ajustados com base nos seus respectivos gradientes de erro. 4º) O processo se repete até que todos os pesos da rede sejam ajustados. A Figura 7 mostra o sentido do fluxo na fase *forward* e na fase *backward*.

Figura 7 – Fluxos das duas fases do treinamento.



Fonte – (BRAGA; CARVALHO; LUDEMIR, 2016).

3.2.3 O Algoritmo *Back-propagation*

O algoritmo mais usado no treinamento de redes MLP é o *back-propagation*, por isso ele merece ser analisado com toda a riqueza de detalhes. Por essa razão, são mostrados a seguir todos os passos do referido algoritmo com todas as fórmulas utilizadas por ele,

de acordo com as explicações dadas por Rosa (2011), Russell e Norvig (2013) e Braga, Carvalho e Ludemir (2016). Vale ressaltar que aqui foi usada como base uma rede MLP com uma camada de entrada indexada pela letra i , uma camada intermediária (escondida) indexada pela letra j e uma camada de saída indexada pela letra k .

Passo 1: Inicialização.

Todos os pesos da rede MLP, aqui representados pela letra W , e os níveis das bias (limiar de ativação), aqui representados pela letra θ , são inicializados randomicamente.

Passo 2: Ativação.

A rede é ativada por meio da aplicação das entradas (*input*) $x_1(p), x_2(p), \dots, x_n(p)$ e saídas desejadas (*target*) $y_{d1}(p), y_{d2}(p), \dots, y_{dn}(p)$. É importante notar que neste algoritmo a letra y sempre representa uma saída e a letra x uma entrada. Em seguida:

a) A saída real de cada neurônio da camada escondida é calculada com a seguinte fórmula: $y_j(p) = \text{sigmoid}[\sum_{i=1}^n X_i(p) \times W_{ij} - \theta_j]$, em que n representa o número de entradas do neurônio j da camada escondida e sigmoid é a função de ativação $f(x) = 1/(1 + e^{-x})$.

b) A saída real de cada neurônio da camada de saída é calculada usando a seguinte fórmula: $y_k(p) = \text{sigmoid}[\sum_{j=1}^m X_{jk}(p) \times W_{jk}(p) - \theta_k]$, em que m representa o número de entradas do neurônio k da camada de saída.

Passo 3: Treinamento dos pesos.

Os pesos da rede MLP são atualizados por meio da propagação *backward* dos erros associados aos neurônios da camada de saída. É importante lembrar que a camada de saída pode ter um ou mais neurônios.

a) O gradiente de erro para cada neurônio da camada de saída é calculado com a fórmula: $\delta_k(p) = y_k(p) \times [1 - y_k(p)] \times e_k(p)$, em que $e_k(p) = y_{dk}(p) - y_k(p)$ e “ $y_k(p) \times [1 - y_k(p)]$ ” é a derivada da função de ativação sigmoideal.

As correções dos pesos de cada neurônio da camada de saída são calculadas por meio da equação: $\Delta w_{jk}(p) = \alpha \times y_j(p) \times \delta_k(p)$, em que α representa a taxa de aprendizagem. Lembrando que a saída y_j da camada escondida é a entrada da camada de saída.

Os pesos de cada neurônio da camada de saída são atualizados usando a equação: $w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p)$.

b) O gradiente de erro para cada neurônio da camada escondida é calculado com a fórmula: $\delta_j(p) = y_j(p) \times [1 - y_j(p)] \times \sum_{i=1}^l \delta X_{jk}(p) \times W_{jk}(p)$, em que l representa o número de neurônios da camada de saída e $\sum_{i=1}^l \delta X_{jk}(p) \times W_{jk}(p)$ representa a somatória ponderada dos gradientes de erro dos neurônios da camada de saída.

As correções dos pesos de cada neurônio da camada escondida são calculadas com a equação: $\Delta w_{ij}(p) = \alpha \times x_i(p) \times \delta_j(p)$.

Os pesos de cada neurônio da camada escondida são atualizados usando a fórmula:
 $W_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p)$.

Passo 4: Iteração.

A iteração p é incrementada de 1, o algoritmo volta para o passo 2 (ativação) e repete todo o processo até que o critério de erro selecionado seja satisfeito.

Normalmente, o algoritmo *back-propagation* chega ao fim quando o erro quadrático atinge um valor mínimo previamente determinado ou se depois de um número determinado de épocas o erro quadrático não diminuir. O erro quadrático é basicamente a média da somatória dos erros obtidos para cada par de saída desejada e saída real elevado ao quadrado, o que é mostrado na Equação (3.7), na qual p é o número de exemplos de treinamento. É importante destacar que o erro quadrático é calculado para o conjunto completo dos dados de treinamento e sempre que o passo 4 percorre o conjunto completo de treinamento uma época é anotada. Sendo assim, o gráfico de convergência da rede deve mostrar a relação época \times erro quadrático.

$$e = \frac{1}{p} \sum_{i=1}^p (y_{dk} - y_k)^2 \quad (3.7)$$

4 Experimentos Realizados

Os experimentos seguintes procuraram prever o consumo de água em função da temperatura, umidade e quantidade de chuva usando RNAs implementadas por meio de ferramentas do Matlab. Diferentes configurações de RNAs e diferentes ferramentas foram postas em teste para se tentar chegar a uma solução otimizada. Um segundo objetivo foi comparar as duas ferramentas que o Matlab dispõe para a implementação de RNAs: *Neural Network Toolbox graphical user interface* (Nntool) e *Neural Network Start GUI* (Nnstart).

4.1 Primeiro Experimento

4.1.1 Pré-processamento dos Dados de Treino e Teste

Os dados do SAA Setor Águas Negras foram repassados pelo desenvolvedor e administrador do sistema Telemetry que, conforme indicado anteriormente, encontra-se instalado no referido setor. Os dados estavam dispostos em uma tabela com quatro colunas: “data”, “hora”, “vazão do poço” e “nível do reservatório”. As linhas da tabela percorriam todo o mês de outubro de 2017, minuto a minuto. Com base nas informações de vazão do poço, nível do reservatório e do conhecimento de um engenheiro sanitarista, foi possível criar um *script* no Matlab para determinar o consumo de água do sistema em m^3/min . Posteriormente, foi desenvolvido um outro *script*, também no Matlab, para encontrar o consumo em m^3/h . Esse último valor serviu como *target* para a RNA. Vale lembrar que *target* é o valor alvo, ou seja, o valor desejado para a saída da rede.

As entradas da RNA foram obtidas no site do Instituto Nacional de Meteorologia (Inmet) na forma de uma tabela contendo os dados de “temperatura”, “umidade” e “precipitação (chuva)” do mês de Outubro de 2017, hora a hora. Em seguida, os dados de entrada e o *target* foram normalizados por escalonamento decimal para ficarem entre os valores 0 e 1. De acordo com Neves (2003), a fórmula para esse tipo de normalização é a que aparece na Equação (4.1), em que j é o menor inteiro tal que $\text{Max}(|v'|) < 1$. Essa normalização é necessária porque 1 é o valor máximo que a função de ativação sigmoideal pode atingir. Para os dados de entrada, j foi igual a 2 e para os valores de *target*, j foi igual a 3.

$$\text{Para um valor } v \text{ de } A, v' = \frac{v}{10^j} \quad (4.1)$$

Por fim, os valores foram postos em uma sequência aleatória usando a função $A = A(\text{randperm}(\text{end}), :)$ do Matlab e finalmente divididos em duas partes. A maior parte

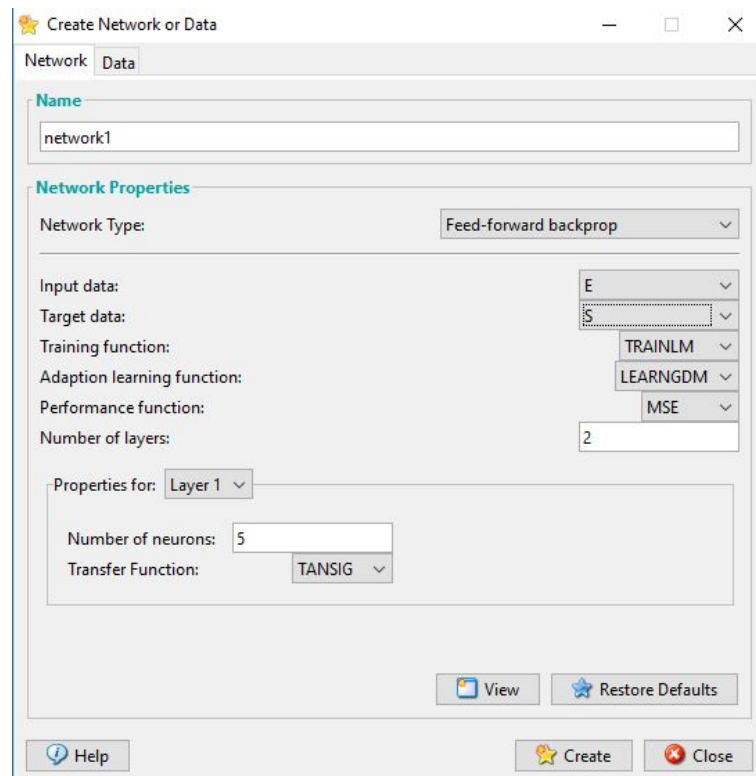
(70%), o que corresponde a 472 instâncias, foi destinada para o treinamento da rede e a outra parte (30%), que corresponde a 200 instâncias, foi destinada para o teste da rede.

É importante notar que para se chegar às matrizes usadas no experimento, isto é, uma matriz 3×472 para os dados de entrada usados no treinamento e uma matriz 3×200 para os dados de entrada usados no teste, foi necessário encontrar as matrizes transpostas da tabela original, usando uma função do Matlab. Essa é a estrutura que os dados devem ter para serem usados no Kit de ferramentas de RNA do Matlab chamado de Nntool.

4.1.2 Arquitetura e Parâmetros da Rede Neural Utilizada

A RNA foi criada usando o *Shell Nntool*¹ do Matlab R2014a, que possui uma interface gráfica com todas as ferramentas necessárias para a criação, treino e teste de uma RNA. A referida rede contou com uma camada de entrada com 3 entradas, uma camada escondida com 5 neurônios e uma camada de saída com 1 neurônio e uma única saída. Na criação optou-se por usar os parâmetros padrões do Matlab porque eles coincidem com sugestões encontradas na literatura, especialmente em Braga, Carvalho e Ludemir (2016) e Santos (2011). Esses parâmetros são mostrados na Figura 8.

Figura 8 – Parâmetros usados na criação da rede neural.

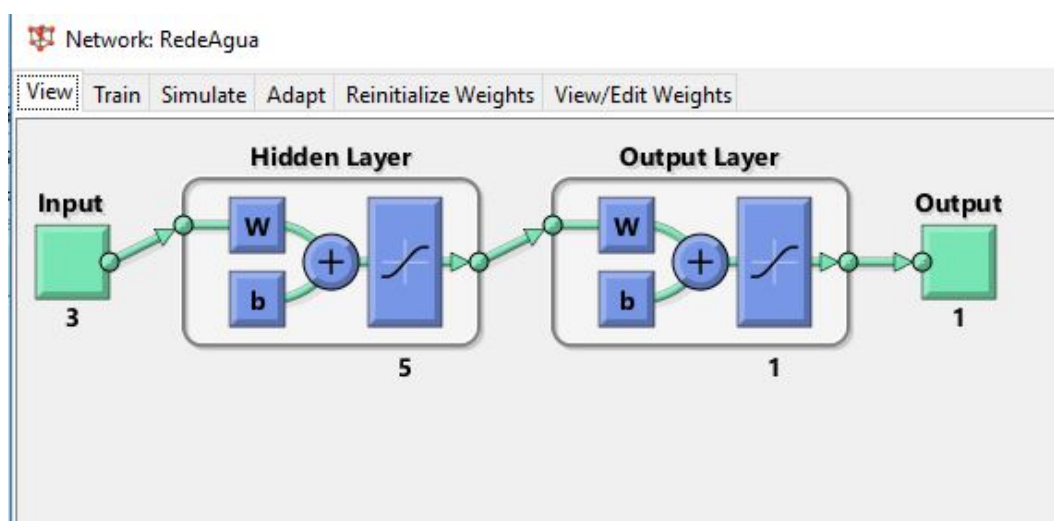


Fonte – Elaborada pelo autor.

¹ A documentação dessa ferramenta encontra-se em <https://www.mathworks.com/help/nnet/ref/nntool.html>.

Vale ressaltar que se trata de uma RNA do tipo *feedforward* que usa o algoritmo *back-propagation*. A expressão TRAINLM, que aparece relacionada ao parâmetro “*training function*” na Figura 8, refere-se a função de treinamento *Levenberg-Marquardt*, que nada mais é do que uma variação do algoritmo *back-propagation* tradicional. A função de ativação (*transfer function*) usada em todos os neurônios é a TANSIG (*Symmetric sigmoid transfer function*). O critério que determina o fim do processamento (*performance function*) é o *Mean Squared Error* (MSE), ou erro quadrático. A arquitetura da rede criada aparece claramente na Figura 9.

Figura 9 – Arquitetura da RNA criada.



Fonte – Elaborada pelo autor.

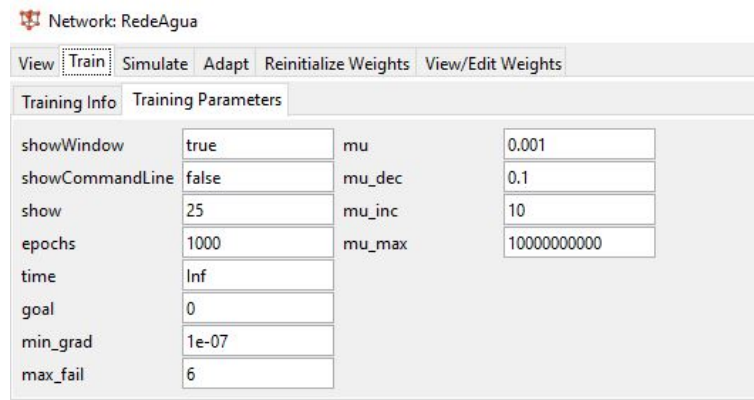
4.1.3 Treino da RNA

Os parâmetros de treinamento utilizados aparecem na Figura 10. De acordo com esses parâmetros o treinamento deve chegar ao fim quando ocorrer uma das seguintes condições: o número máximo de 1000 épocas for atingido; a soma do erro quadrático for igual a 0; o gradiente de erro for menor que 0,0000001; o erro quadrático não diminuir após 6 épocas; ou “mu” for maior que 10000000000. O “mu” é um fator adaptativo, ao qual é adicionado “mu_inc” até que o erro quadrático diminua. A mudança é então feita para a rede e “mu_dec” é subtraído de “mu”.

O resultado do treinamento é apresentado na Figura 11 e o gráfico de convergência do erro quadrático é apresentado na Figura 12. Confrontando as informações contidas nas duas figuras, pode-se afirmar que o treinamento parou na 9ª época porque o erro quadrático não diminuiu após 6 épocas consecutivas, o que é um comportamento típico. Essa condição de parada está destacada com a cor verde na Figura 11. Isso significa que o erro quadrático começou em 0,0351, atingiu o seu menor nível na 3ª época, quando foi igual a 0,0019261, como mostra a Figura 12, e depois permaneceu praticamente estável

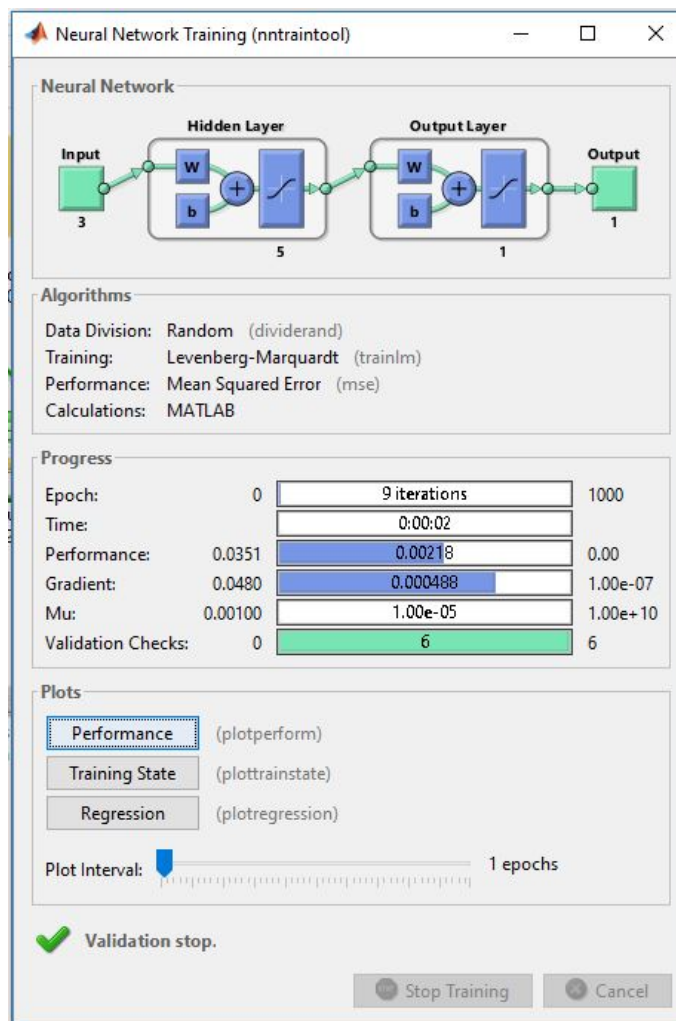
atingindo 0,00218 na 9ª e última época.

Figura 10 – Parâmetros de treinamento.



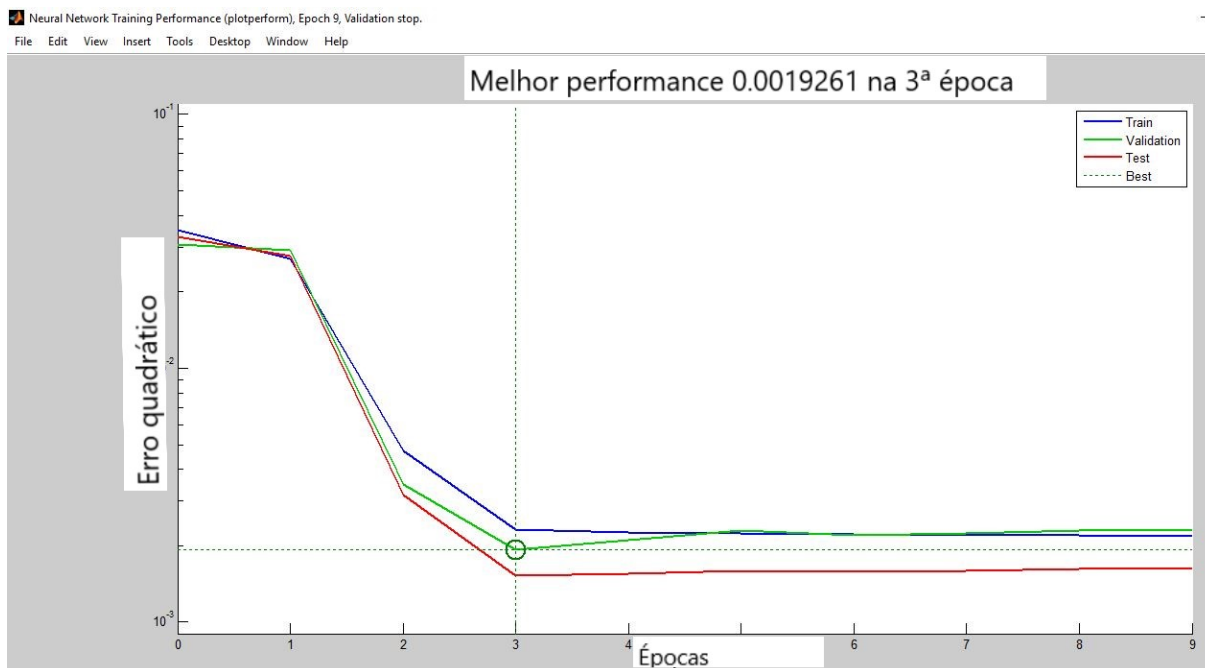
Fonte – Elaborada pelo autor.

Figura 11 – Resultado do treinamento da rede neural.



Fonte – Elaborada pelo autor.

Figura 12 – Convergência do erro quadrático.



Fonte – Elaborada pelo autor.

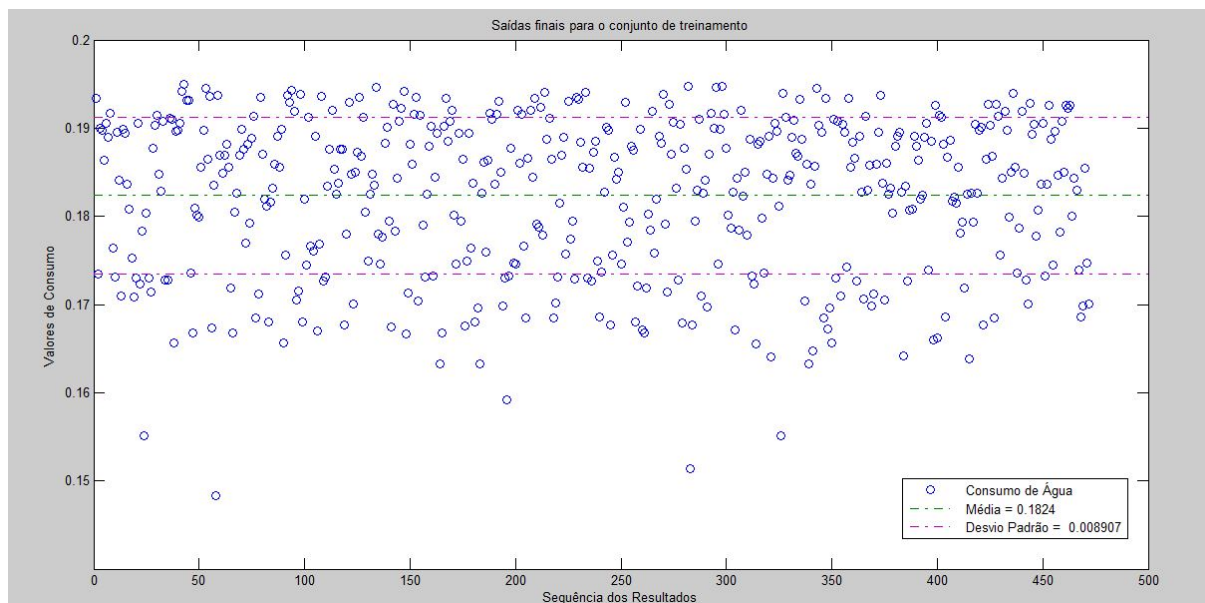
É preciso destacar que o gráfico da convergência do erro quadrático apresenta três curvas e não apenas uma. Isso acontece porque, segundo Beale, Hagan e Demuth (2014) autor da documentação do Matlab R2014a, o algoritmo de treinamento *Trainlm* cria automaticamente um vetor de teste e um vetor de validação separando aleatoriamente uma parte do conjunto de treino. O vetor de validação é usado para o monitoramento da condição de parada que determina o fim do treino caso a soma do erro quadrático não melhore em 6 épocas consecutivas. O vetor de teste é usado como uma verificação adicional de que a rede está generalizando bem, mas não tem nenhum efeito no treinamento.

A Figura 13 mostra as saídas finais obtidas na fase de treinamento da RNA. Os demais resultados, tais como: pesos finais da camada escondida, pesos finais da camada de saída, *bias* finais da camada escondida e *bias* final da camada de saída foram omitidos por não serem relevantes para o presente estudo.

Em relação à Figura 13 é importante observar que o conjunto dos dados obtidos possui um desvio padrão pequeno, o que significa, segundo Morettin e Bussab (2012), que os dados não se afastam muito da média. Pode-se afirmar também que a distribuição dos resultados de consumo de água se aproxima de uma distribuição normal conforme pode ser visto na Figura 14, que mostra a distribuição de frequência dos resultados (histograma) com um formato parecido com a curva de sino característica da distribuição normal. Como em todo histograma, a base de cada retângulo (eixo X) representa uma classe (uma das faixas de valores em que os dados de consumo foram previamente ordenados). A altura de

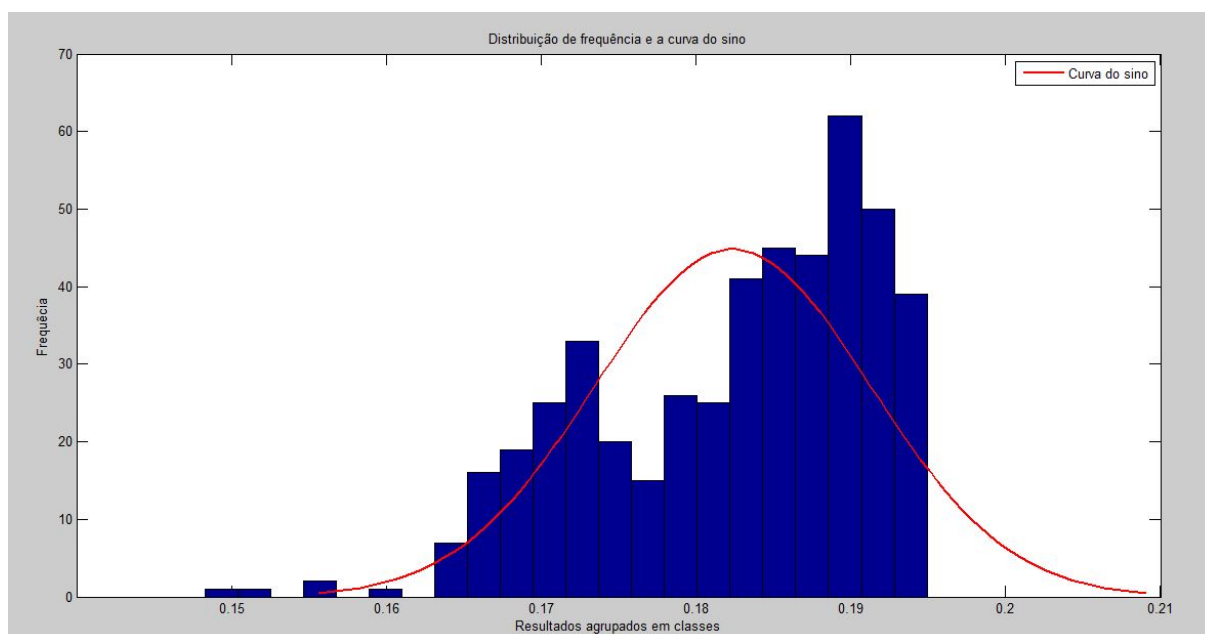
cada retângulo (eixo Y) representa a quantidade ou a frequência com que o valor da classe ocorre no conjunto de dados.

Figura 13 – Saídas finais obtidas.



Fonte – Elaborada pelo autor.

Figura 14 – Histograma com os valores de saída obtidos.

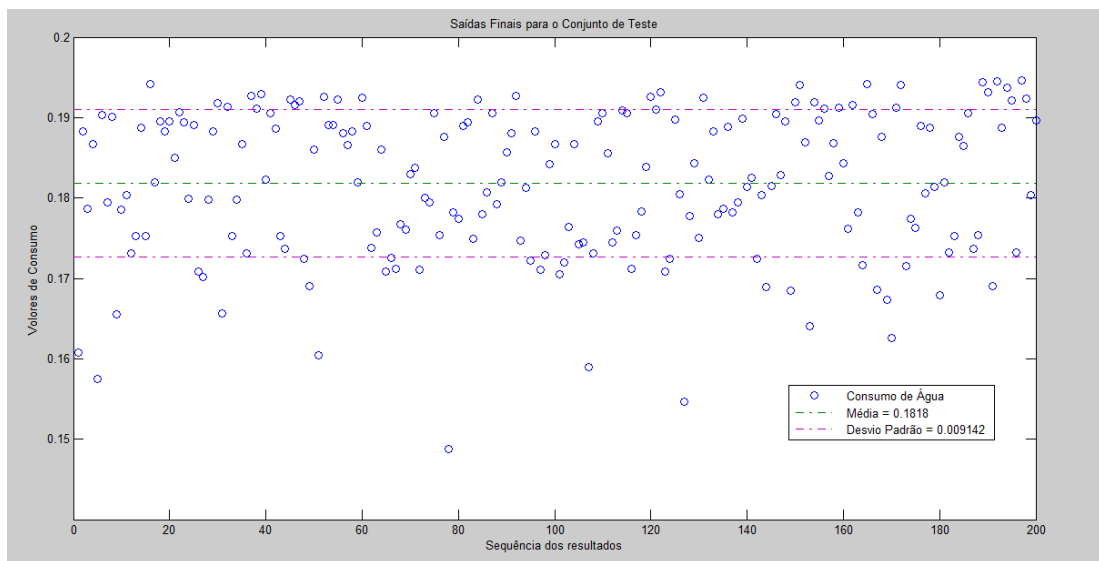


Fonte – Elaborada pelo autor.

4.1.4 Execução da RNA Com os Dados de Teste

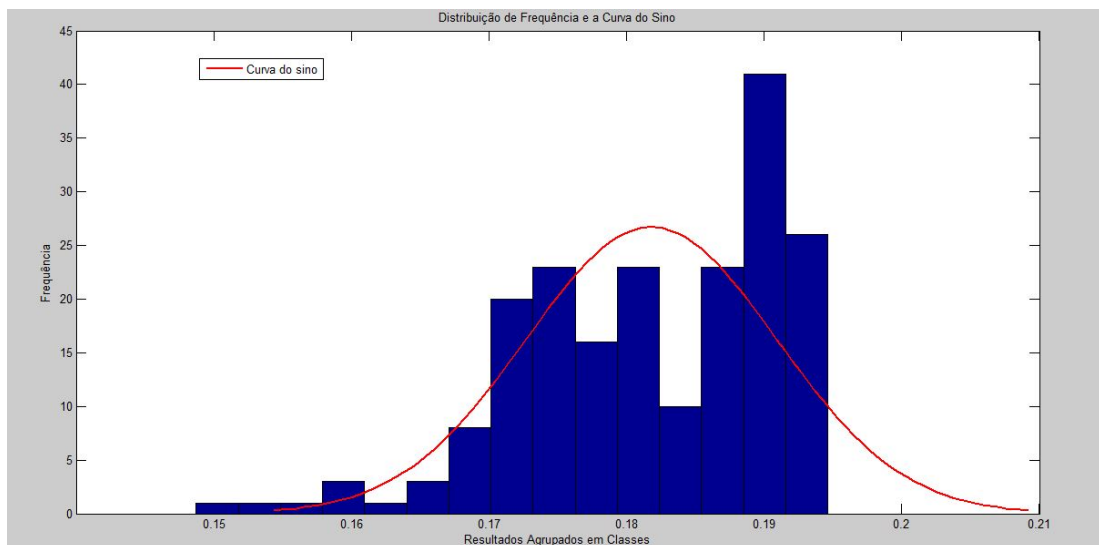
Conforme explicação anterior, uma base de teste foi separada previamente para fazer o teste da RNA após o treinamento. Essa base de teste consiste em uma matriz 3X200 que foi carregada na área de trabalho do Matlab e em seguida introduzida como entrada da RNA. O resultado do teste é mostrado na Figura 15. Note-se que a distribuição dos resultados do teste é bem parecida com a distribuição dos resultados do treinamento, com média e desvio padrão bem próximos.

Figura 15 – Resultado do teste.



Fonte – Elaborada pelo autor.

Figura 16 – Histograma com os valores obtidos no teste.



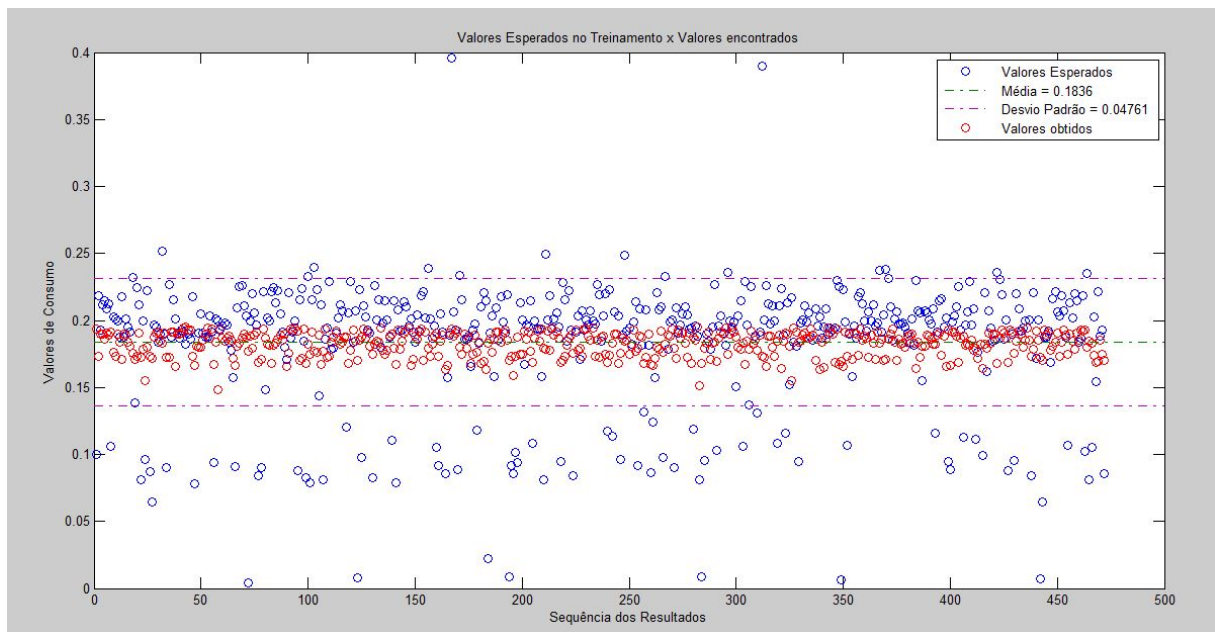
Fonte – Elaborada pelo autor.

A distribuição dos resultados de consumo de água da fase de teste também se aproxima de uma distribuição normal, como mostra a Figura 15. As observações feitas sobre as características do histograma da Figura 14 também valem para a Figura 16. resultado do teste.

4.1.5 Análise dos Resultados Obtidos no Treinamento e no Teste

A Figura 17 confronta as saídas esperadas com as saídas obtidas durante o treinamento. Nela os valores indicados para a média e o desvio padrão são dos valores esperados. Sendo assim é possível observar que os valores obtidos estão dentro do intervalo de um desvio padrão para mais e um desvio padrão para menos dos valores esperados. Logo, a RNA implementada fez uma boa generalização com base nos dados apresentados no treinamento, embora não tenha se ajustado a todos eles. Isso porque, segundo o Teorema Limite Central “Quando o tamanho da amostra aumenta, independentemente da forma da distribuição da população, a distribuição amostral de X aproxima-se cada vez mais de uma distribuição normal” (MORETTIN; BUSSAB, 2012, p. 279) e em uma distribuição normal a maior parte dos dados se encontra dentro do intervalo de um desvio padrão para mais e um desvio padrão para menos.

Figura 17 – Análise do treinamento.

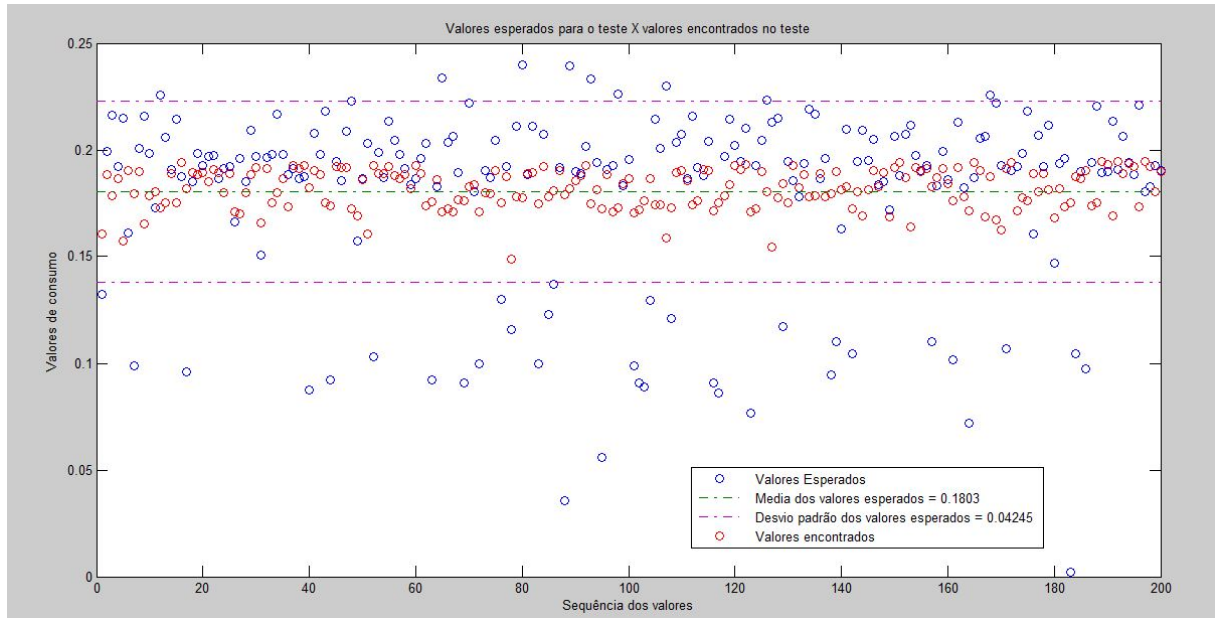


Fonte – Elaborada pelo autor.

Os resultados do teste mostraram uma tendência semelhante à encontrada na fase de treinamento, isto é, os resultados obtido estão dentro do intervalo de um desvio padrão

para mais e um desvio padrão para menos dos resultados esperados. Essa tendência pode ser observada na Figura 18 que confronta as saídas esperadas para o teste e as encontradas.

Figura 18 – Análise do teste.



Fonte – Elaborada pelo autor.

4.2 Segundo Experimento

4.2.1 Pré-processamento dos Dados de Treino e Teste

No segundo experimento foi usada a mesma base de dados do primeiro, com a diferença de que no segundo experimento a base não foi dividida em duas partes (treino e teste). Isso porque a ferramenta do Matlab usada no segundo teste faz essa separação automaticamente e divide a base de dados em duas partes aleatoriamente escolhidas. Dessa maneira, a entrada da RNA foi uma matriz 3×672 e o *target* foi um vetor 1×672 .

4.2.2 Arquitetura e Parâmetros da Rede Neural Utilizada

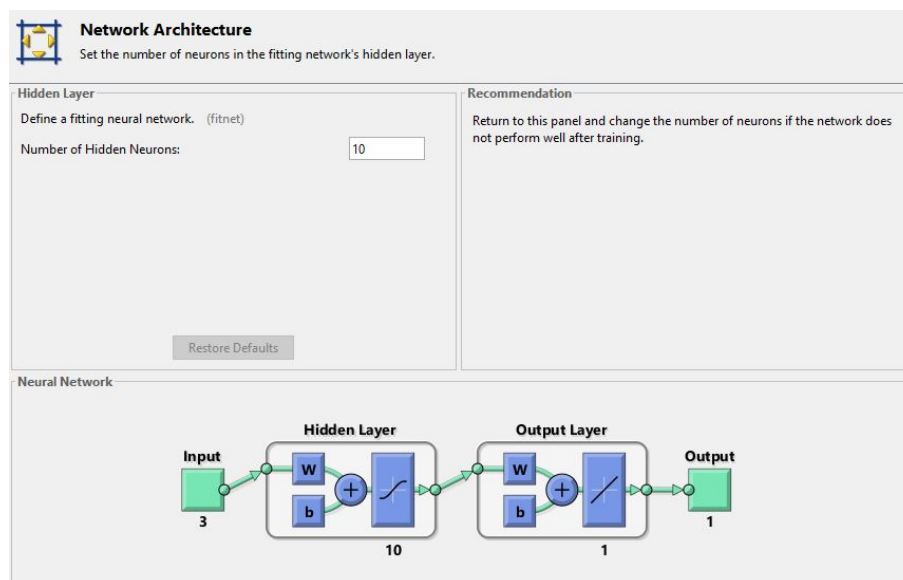
No segundo teste, optou-se por usar o pacote Nnstart² do Matlab R2014a. Esse pacote também possui interface gráfica com todas as ferramentas necessárias para criar, treinar e testar uma RNA. Contudo, o pacote Nnstart é mais moderno que o Nntool e, como tal, possui mais recursos e facilidades. O Nnstart também possui uma interface mais amigável e automatiza várias funções que no Nntool precisavam ser feitas manualmente. Na tela inicial do Nnstart, o usuário escolhe que tipo de RNA quer criar, de acordo com a

² A documentação dessa ferramenta encontra-se em <https://www.mathworks.com/help/nnet/ref/nnstart.html>.

finalidade da rede a ser criada. Neste experimento, a opção foi por regressão não-linear que o Matlab chama de *curve fitting* (ajuste de curva).

A RNA construída foi semelhante a usada no primeiro experimento, isto é, uma RNA *feed-forward*, com uma camada de entrada, uma camada escondida e uma camada de saída. O algoritmo de treinamento também foi o mesmo: *Levenberg-Marquardt backpropagation*. A diferença recaiu apenas na função de ativação. No primeiro experimento foi usada uma função de ativação sigmoideal tanto na camada escondida, como na camada de saída. Neste segundo experimento, foi usada uma função de ativação sigmoideal na camada escondida e uma função de ativação linear na camada de saída. A referida configuração aparece com frequência na literatura. Pode ser vista como alternativa, por exemplo, em Braga, Carvalho e Ludemir (2016) e na própria documentação do Matlab R2014a.

Figura 19 – Arquitetura da 2ª RNA.



Fonte – Elaborada pelo autor.

O passo seguinte foi a seleção dos dados de *input* e *target*, que foram descritos no tópico anterior e já se encontravam na área de trabalho do Matlab. Seguiu-se então, o passo decisivo: escolher a quantidade de neurônios da camada escondida. Como já foi comentado no presente trabalho, não existe nenhuma regra, teórica ou prática, que determine de forma clara e objetiva a quantidade exata de neurônios que uma camada escondida deve ter. Sendo assim, a quantidade de neurônios mais apropriada deve ser encontrada pelo método das tentativas. Como no experimento anterior foram usados 5 neurônios na camada escondida, no segundo experimento, optou-se por usar 10 neurônios na camada escondida. O objetivo era chegar a uma RNA com uma complexidade média capaz de se ajustar bem aos dados de treino e, ao mesmo tempo, criar uma boa generalização para se ajustar aos dados de teste. A seleção do número de neurônios e a arquitetura da RNA aparecem na

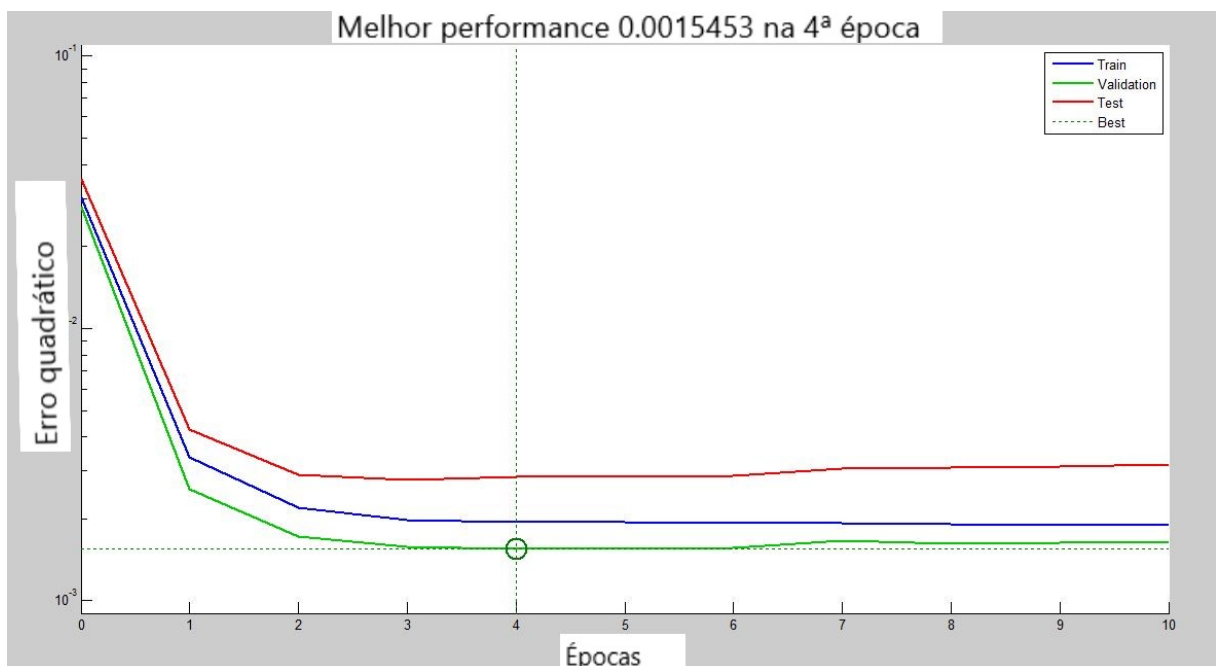
Figura 19. Nela é possível ver uma RNA com três entradas na camada de entrada, uma camada escondida com 10 neurônios com função de ativação sigmoideal e uma camada de saída com uma saída, um neurônio e função de ativação linear.

4.2.3 Treino da RNA e Resultados

Antes de iniciar o treino da RNA, o Nnstart dividiu randomicamente a base de dados nas seguintes partes: 70 % (470 exemplos) para o treino, 15 % (101 exemplos) para a validação e 15 % (101 exemplos) para o teste. A função de cada uma dessas partes já foi explicada no primeiro experimento e continua a mesma.

A Figura 20 mostra o gráfico com a convergência do erro quadrático após o treino. É importante notar que a rede convergiu para o menor erro quadrático na 4ª época e parou na 10ª época, após 6 épocas sem melhorar a performance. A RNA do 1º experimento atingiu a melhor performance na 3ª época, porém a melhor performance da 1ª RNA foi 0,0019261, enquanto que a melhor performance da 2ª rede foi 0,0015453. Sendo assim, pode-se dizer que a 2ª RNA se ajustou melhor aos dados de treinamento. Contudo, a curva correspondente aos dados de teste se afastou bastante da linha ideal atingindo um erro quadrático de 0,0028430. Isso indica que a rede está superdimensionada o que a torna capaz de se ajustar melhor aos dados de treino, mas não a tornar capaz de fazer uma generalização melhor para poder se ajustar aos novos dados apresentados no teste. Esse fenômeno é conhecido como *overfitting*.

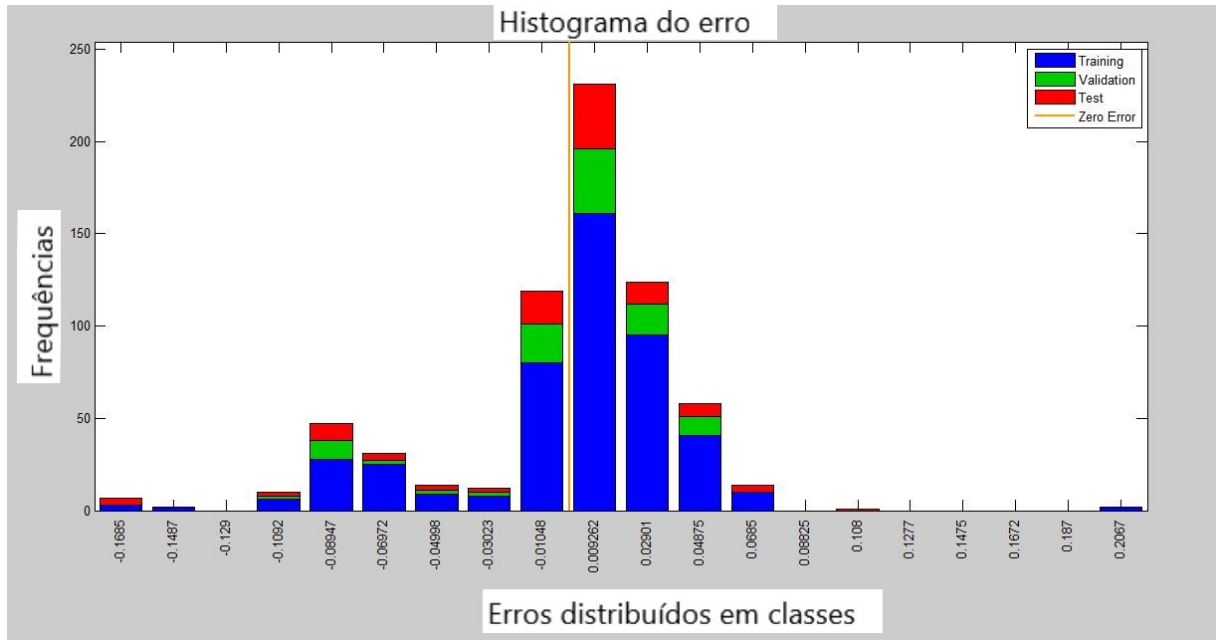
Figura 20 – Convergência do erro quadrático para o 2º experimento.



Fonte – Elaborada pelo autor.

O histograma do erro (saída deseja – saída obtida) mostra que erros maiores foram mais frequentes durante a fase de teste, o que corrobora a explicação dada acima. Esse histograma aparece na Figura 21.

Figura 21 – Histograma do erro do 2º experimento.



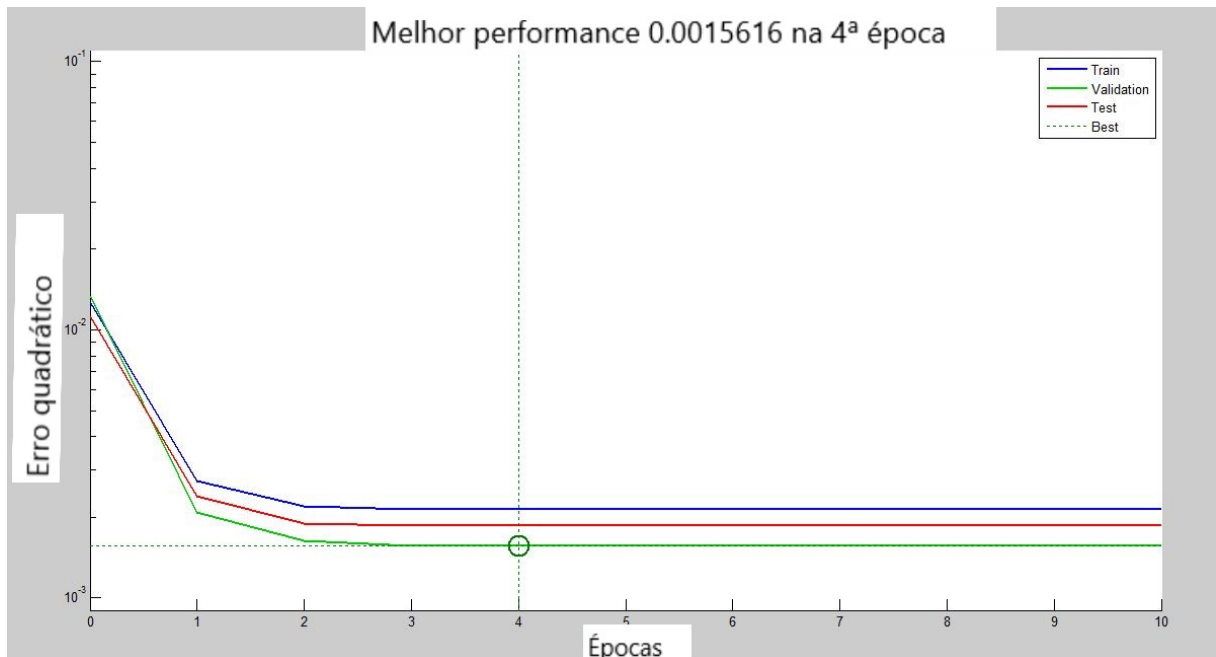
Fonte – Elaborada pelo autor.

4.3 Terceiro Experimento

Em um terceiro momento, foi feito um outro experimento usando a mesma ferramenta do segundo experimento, com a mesma configuração de RNA, mesma arquitetura e mesmo algoritmo de treinamento. A única mudança feita foi colocar 5 neurônios na camada escondida. O resultado desse último treino é mostrado da Figura 22.

A Figura 22 mostra que a arquitetura e a configuração da RNA do terceiro experimento é melhor do que a do primeiro experimento porque atingiu um erro quadrático menor durante o treino (0,0015616). A RNA do terceiro experimento também é melhor do que a do segundo experimento porque permanece com um erro quadrático pequeno durante a fase de teste (0,00185745).

Figura 22 – Convergência do erro quadrático para o 3º experimento.



Fonte – Elaborada pelo autor.

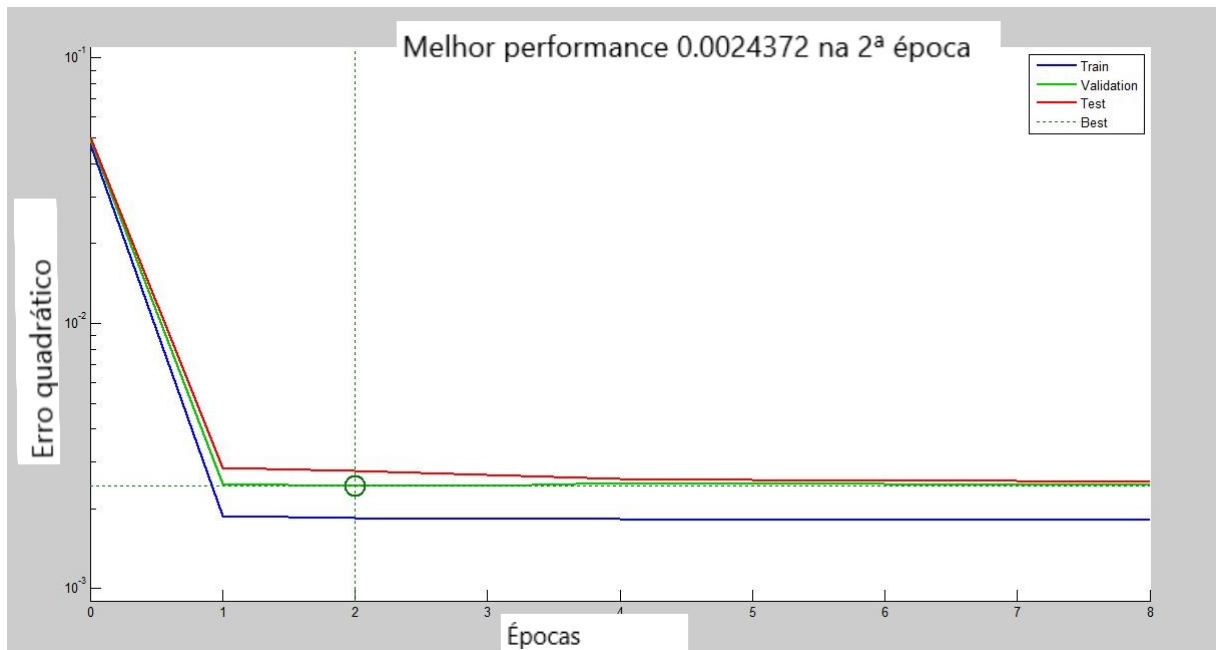
4.4 Quarto Experimento

Para concluir, foi feito um último experimento, também usando a mesma ferramenta, a mesma configuração de RNA, a mesma arquitetura de rede e o mesmo algoritmo de treinamento do segundo experimento. O objetivo desse último experimento era saber se não era possível simplificar a rede ainda mais, usando 3 neurônios na camada escondido. Isso porque para a abordagem multiobjetivo, explicada no tópico que tratou de “arquitetura de redes MLP”, é preciso buscar um RNA ao mesmo tempo com um erro mínimo e uma estrutura mínima.

A resposta foi negativa porque a última rede apresentou erro quadrático maior que todas as outras no treino (0,0024372) e o segundo pior no teste (0,0027688). Esse resultado pode ser visto na Figura 23.

A Tabela 1 resume todos os resultados obtidos nos quatro experimentos (Exp) e mostra claramente que a RNA usada no terceiro experimento é a melhor porque obteve menor média entre o erro quadrático do treinamento e do teste.

Figura 23 – Convergência do erro quadrático para o 4º experimento.



Fonte – Elaborada pelo autor.

Tabela 1 – Resumo dos resultados dos experimentos.

	Função de ativação na saída	Nº de neurônios na camada escondida	Quantidade de épocas que convergiu	Erro quadrático do treinamento	Erro quadrático do teste	Média dos erros quadráticos (treinamento e teste)
Exp 1	Sigmoidal	5	9	0,0019	0,0018	0,0018
Exp 2	Linear	10	10	0,0015	0,0028	0,0021
Exp 3	Linear	5	10	0,0015	0,0018	0,0017
Exp 4	Linear	3	8	0,0024	0,0027	0,0026

5 Considerações Finais e Trabalhos Futuros

Após quatro experimentos chegou-se a conclusão de que a melhor arquitetura e configuração de RNA para resolver o problema proposto é uma rede com uma única camada escondida com 5 neurônios e função de ativação sigmoidal, e uma camada de saída com um neurônio e função de ativação linear (rede do experimento 3). Esse resultado coincide com o trabalho feito por Santos (2011) que usou uma RNA com uma camada escondida com 5 neurônios para resolver um problema semelhante. Também coincide com as recomendações feitas por Braga, Carvalho e Ludemir (2016) e pela documentação do Matlab R2014a que aconselham o uso de uma função de ativação linear na camada de saída.

Pode-se afirmar que o presente trabalho atingiu o seu principal objetivo que era implementar uma RNA capaz de prever o consumo de água em função de variáveis climatológicas. Todos os objetivos específicos também foram atingidos, quais sejam: levantar dados climatológicos (temperatura, umidade e chuva) e de consumo de água referentes ao Setor Águas Negras; realizar o pré-processamento dos dados e gerar matrizes com as variáveis escolhidas para serem usadas na rede neural artificial; elaborar redes neurais artificiais, com diferentes configurações, usando o Matlab para prever a demanda futura por água no referido setor; e analisar os resultados obtidos, identificando a arquitetura de rede neural mais adequada para o problema em questão. Além disso, a motivação de estimular o uso de Inteligência Artificial na análise e no planejamento de SAA também foi contemplada.

A principal dificuldade deste trabalho foi conseguir uma base de dados que pudesse ser usada de forma satisfatória nos experimentos. Mas essa dificuldade é inerente ao setor de abastecimento de água que não investe na coleta de dados porque também não tem uma tradição de análise e planejamento. Sendo assim, este trabalho também deu uma pequena contribuição para quebrar esse círculo vicioso, mostrando a importância de coletar dados significativos para serem usados em sistemas inteligentes de planejamento e gestão.

Em um futuro próximo, o autor deste trabalho pretende investir em uma rede de sensores para coletar dados na área de saneamento e meio ambiente e oferecer o serviço de análise de dados e ciência de dados em parceria com uma empresa do setor ambiental.

A combinação de uma base maior de dados coletadas por sensores e a introdução de variáveis socioeconômicas deve trazer maior rigor científico para experimentos futuros, além de abrir novas possibilidades para uma compreensão mais abrangente dessa área vital que é o gerenciamento dos recursos hídricos.

Pretende-se também utilizar outras técnicas ou combinações de técnicas de Inteli-

gência Artificial na busca por previsões de consumo mais precisas, por exemplo RNA com algoritmos genéticos.

Referências

- BEALE, M. H.; HAGAN, M. T.; DEMUTH, H. B. *Neural Network Toolbox: User's Guide. Natick: MathWorks*,. 2014. Citado na página 38.
- BRAGA, A. d. P.; CARVALHO, A. P. d. L. F.; LUDEMIR, T. B. *Redes Neurais Artificiais: Teoria e Aplicação*. [S.l.: s.n.], 2016. Citado 10 vezes nas páginas 24, 25, 26, 27, 29, 31, 32, 35, 43 e 48.
- COSANPA. *Sistema de Abastecimento de Água da Região Metropolitana de Belém, Síntese do plano diretor*. 2006. Citado 2 vezes nas páginas 18 e 19.
- GOMES, H. P. *Sistema de Bombeamento: Eficiência Energética*. 2009. UFPB. Editora Universitária. João Pessoa. Citado na página 21.
- IBGE. *Estimativas de População*. 2015. Disponível em: <ftp://ftp.ibge.gov.br/Estimativas_de_Populacao/Estimativas_2015/estimativa_TCU_2015_20170614>. Citado na página 16.
- IGLÉSIAS, P.; GOMES, H.; GARCÍA, R. *Abastecimento de Água: o estado da arte e técnicas avançadas*. 2007. UFPB. Editora Universitária. Citado na página 17.
- INMET. *Instituto Nacional de Meteorologia*. 2017. Disponível em: <<http://www.inmet.gov.br/portal/index.php?r=tempo/graficos>> Citado na página 22.
- MORETTIN, P. A.; BUSSAB, W. d. O. *Estatística Básica*. [S.l.]: Saraiva, 2012. Citado 2 vezes nas páginas 38 e 41.
- NEVES, R. d. C. D. *Pré-processamento no Processo de Descoberta de Conhecimento em Banco de Dados*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, 2003. Citado na página 34.
- PLANSAB. 2013. Disponível em: <http://www.mma.gov.br/port/conama/processos/AECBF8E2/Plansab_Versao_Conselhos_Nacionais_020520131.pdf> Citado na página 18.
- PNSB. 2008. Citado na página 18.
- ROSA, G. *Fundamentos da Inteligência Artificial*. [S.l.]: LTC, 2011. Citado na página 32.
- RUSSELL, S.; NORVIG, P. *Inteligência Artificial*. [S.l.]: Elsevier, 2013. Citado 2 vezes nas páginas 23 e 32.
- SANTOS, C. C. d. *Previsão de Demanda de Água na Região Metropolitana de São Paulo com Redes Neurais Artificiais e Condições Socioambientais e Meteorológicas*. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo, 2011. Citado 2 vezes nas páginas 35 e 48.
- TSUTIYA, M. T. *Abastecimento de Água*. [S.l.]: Escola Politécnica de São Paulo, 2006. Citado na página 15.

Estatística Municipal. Disponível em: <<http://fapespa2.pa.gov.br/index.php/estatistica-municipal>.> Acesso em: 05/02/2018 Nenhuma citação no texto.

SIZEMORE, J.; MUELLER, P. Matlab para Leigos. Rio de Janeiro: Alta Books, 2016. Nenhuma citação no texto.