



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA QUÍMICA
TRABALHO DE CONCLUSÃO DE CURSO EM ENGENHARIA
QUÍMICA



Comparação de métodos de regularização no treinamento de redes neurais artificiais aplicado a uma torre de resfriamento de água

Autor: Artur Tiburski Vaz Costa

Orientador: Pedro Rafael Bolognese Fernandes

Porto Alegre, outubro de 2022

Autor: Artur Tiburski Vaz Costa

Comparação de métodos de regularização no
treinamento de redes neurais artificiais aplicado a uma
torre de resfriamento de água

*Trabalho de Conclusão de Curso apresentado à
COMGRAD/ENQ da Universidade Federal do Rio
Grande do Sul como parte dos requisitos para a
obtenção do título de Bacharel em Engenharia
Química*

Orientador: Pedro Rafael Bolognese Fernandes

Banca Examinadora:

Prof. Dr. Marcio Schwaab, UFRGS

Dr. Jônathan William Vergani Dambros, UFRGS

Porto Alegre

2022

AGRADECIMENTOS

Meus agradecimentos são dedicados a todos que me auxiliaram, de qualquer forma, a chegar até este ponto em minha vida e vencer esta etapa: meu pai, Augusto, minha mãe, Rosa, minha irmã, Bianca, meu sobrinho, Oliver, minha namorada e companheira de vida, Mariana, e meus amigos de infância, do trabalho, da faculdade e de vivências do passado. Também agradeço ao grande Prof. Pedro Fernandes que, mesmo durante estes tempos complexos e remotos, forneceu apoio técnico, de companheirismo e, com sua paciência e orientação, me auxiliou imensamente no desenvolvimento deste trabalho. Agradeço também à M.E. Lara Brum pelo auxílio na obtenção dos dados do processo utilizado. O caminho até chegar nesta etapa foi árduo, mas com a presença e incentivos de todas essas pessoas foi possível, enfim, vencer.

RESUMO

O crescente desenvolvimento e aplicação de técnicas de inteligência artificial a processos nas mais diversas indústrias promove a necessidade de, cada vez mais, estudar e adaptar o uso destas ferramentas a processos da indústria química. Entre estas ferramentas estão as Redes Neurais Artificiais, cuja capacidade de generalização tem o potencial de, sob as corretas condições da base de dados e treinamento, proporcionar resultados muito próximos da realidade. Nesse sentido, este trabalho tem como objetivo testar as técnicas de regularização comumente empregadas no treinamento de redes neuronais, as quais visam evitar os problemas de sobre-ajuste (*overfitting*) associados ao modelo, tendo como base o estudo de caso de uma torre industrial de resfriamento de água. As técnicas testadas foram as regularizações L1, L2 e *Dropout*, que foram aplicadas a redes treinadas em diversas configurações e com a utilização de variados valores de hiperparâmetros. Através da análise dos resultados das métricas de desempenho erro médio absoluto (MAE) e erro quadrático médio (RMSE), assim como do comportamento das curvas de aprendizado resultantes do treinamento das redes sob diversas condições, foi possível verificar quais arquiteturas de rede obtiveram os melhores resultados de predição sem regularização de dados, para os casos de uma e duas camadas ocultas. A técnica L2 com $\lambda = 0,01$ no caso da RNA com 5 nós internos foi o único caso em que a regularização resultou em valores de MAE e RMSE menores que a rede sem regularização. A partir deste resultado, através de análise exploratória, foi investigado o valor de λ (L2) que proporcionasse os menores MAE e RMSE para esta rede, chegando ao valor de $\lambda = 0,01597$. Contudo, foi verificado que, de forma geral, as técnicas de regularização não proporcionaram melhorias significativas na capacidade de generalização das RNA e não demonstraram, nas curvas de aprendizado, diminuição dos efeitos do sobre-ajuste, mesmo que presentes em pequena magnitude.

Palavras-chave: *inteligência artificial, redes neurais artificiais, regularização de dados, overfitting*

ABSTRACT

The growing development and application of artificial intelligence in processes on various industries promotes the need to increasingly study and adapt the use of these tools to processes of the chemical industry. Among these tools, are the Artificial Neural Networks, which generalization capacities have the potential to, under the correct circumstances of the database and training, deliver results very close to reality. In that regard, this study has, as objective, testing the regularization techniques usually employed in the training of neural networks, which aim at avoiding the problems of overfitting associated to the model, having as study base the analysis of a water industrial cooling tower. The tested techniques were the L1, L2 and Dropout regularizations, which were applied to networks that were trained using various configurations, along with several values of hyperparameters. By analyzing the results of the performance metrics mean absolute error (MAE) and root mean square error (RMSE), as well as the behaviour of the learning curves resulted from the networks' training under various conditions, it was possible to verify which network architectures obtained the best prediction results without data regularization, for the cases of one and two hidden layers. The L2 technique with $\lambda = 0,01$ in the case of the neural network with 5 internal nodes was the only case in which the regularization resulted in values of MAE and RMSE smaller than the network with no regularization. From this result, by using exploratory analysis, it was investigated the value of λ (L2) that provided the smallest MAE and RMSE for this network, obtaining the value of $\lambda = 0,01597$. However, it was verified that, in general, the regularization techniques did not deliver significant improvements in the generalization capacity of the neural networks and did not demonstrate, in the learning curves, a decrease of the overfitting effects, even if present in small intensity.

Keywords: *artificial intelligence, artificial neural networks, data regularization, overfitting*

LISTA DE FIGURAS

Figura 1: Classificações de algoritmos em <i>Machine Learning</i>	3
Figura 2: Representação gráfica dos resultados de técnicas de regressão (esquerda) e de classificação (direita).	4
Figura 3: Estruturas de um único neurônio (nó) em uma Rede Neural Artificial.....	5
Figura 4: Rede Neural Artificial tipo <i>feedforward</i> de múltiplas camadas constituída de dez neurônios de entrada, quatro neurônios ocultos e dois neurônios de saída.	6
Figura 5: Representação das funções de ativação ReLU e Linear	8
Figura 6: Representação de sub-ajuste, sobre-ajuste e um ajuste ideal a dados, respectivamente.....	10
Figura 7: Representação qualitativa de uma curva de aprendizado com dados de treinamento e de validação.....	11
Figura 8: Resultado da clusterização do conjunto de dados original, com base no algoritmo <i>K-Means</i> , utilizando a temperatura de saída da água (°C) e a temperatura de bulbo seco do ar (°C).	14
Figura 9: Etapas de experimento e análise realizadas no estudo.	16
Figura 10: Curvas de aprendizado obtidas no treinamento da RNA com duas camadas ocultas compostas por 4 e 3 nós. À esquerda, a curva associada à função perda (MAE) e à direita a curva associada à métrica RMSE	17
Figura 11: Comparação entre a temperatura de saída da água (°C) real e obtida com a predição através da RNA com uma camada oculta de 5 nós internos para dados de teste....	18
Figura 12: Comparação entre a temperatura de saída da água (°C) real e obtida com a predição através da RNA com duas camadas ocultas de 5 e 4 nós internos para dados de teste.....	19
Figura 13: Curvas de aprendizado obtidas no treinamento, sem regularização, da RNA com duas camadas ocultas com 5 e 4 nós. À esquerda, a curva associada à função perda (MAE) e à direita a curva associada à métrica RMSE	21
Figura 14: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 e 4 nós ocultos utilizando a técnica de regularização de dados L1 com $\lambda = 0,3$	21
Figura 15: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 e 4 nós ocultos utilizando a técnica de regularização de dados L2 com $\lambda = 0,3$	22
Figura 16: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 e 4 nós ocultos utilizando a técnica de regularização de dados <i>Dropout</i> com $p = 0,3$	22
Figura 17: Curvas de aprendizado obtidas no treinamento, sem regularização, da RNA com uma camada oculta com 5 nós. À esquerda, a curva associada à função perda (MAE) e à direita a curva associada à métrica RMSE.....	23

Figura 18: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 nós ocultos utilizando a técnica de regularização de dados L1 com $\lambda = 0,01$	23
Figura 19: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 nós ocultos utilizando a técnica de regularização de dados L2 com $\lambda = 0,01$	24
Figura 20: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 nós ocultos utilizando a técnica de regularização de dados <i>Dropout</i> com $p = 0,1$	24
Figura 21: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 nós ocultos utilizando a técnica de regularização de dados L2 com $\lambda = 0,01597$	26
Figura 22: Comparação entre a temperatura de saída da água (°C) real e obtida com a predição através da RNA (5 nós ocultos, regularização L2 com $\lambda = 0,01597$) para dados de teste	26

LISTA DE TABELAS

Tabela 1: Detalhamento das variáveis analisadas no processo.	13
Tabela 2: Resultados estatísticos da aplicação das RNA treinadas aos dados de teste.....	18
Tabela 3: Resultados obtidos em cada RNA com a aplicação das técnicas de regularização de dados L1, L2 e <i>Dropout</i>	20
Tabela 4: Resultados da análise exploratória para o valor do hiperparâmetro λ , utilizando a técnica de regularização L2, na região próxima a $\lambda = 0,01$	25

SUMÁRIO

1	Introdução	1
2	Revisão Bibliográfica	2
2.1	Fundamentos de <i>Machine Learning</i> , <i>Deep Learning</i> e Inteligência Artificial	2
2.1.1	Inteligência Artificial	2
2.1.2	Machine Learning e Deep Learning	2
2.1.3	Classificações de algoritmos em Machine Learning	2
2.2	Redes Neurais Artificiais (RNA)	4
2.2.1	Fundamentos sobre Redes Neurais Artificiais	4
2.2.2	Estruturas de Redes Neurais Artificiais	4
2.2.3	O processo de aprendizado nas Redes Neurais Artificiais	7
2.2.4	Trabalhos com aplicação de Redes Neurais Artificiais em modelos de processo	9
2.3	Regularização de Dados	9
2.3.1	Sobre-ajuste e Sub-ajuste	9
2.3.2	Curvas de aprendizado	10
2.3.3	Técnicas de Regularização de Dados	11
2.3.4	Trabalhos de aplicação de Rede Neurais Artificiais com Regularização de Dados	12
3	Materiais e Métodos	13
3.1	<i>Softwares</i>	13
3.2	Base de Dados	13
3.3	Procedimento de definição dos dados de treinamento, teste e validação	14
3.4	Definição da estrutura da RNA e seu treinamento	14
3.4.1	Aplicação de técnicas de regularização de dados	16
4	Resultados	17
4.1	Resultados das RNA sem regularização de dados	17
4.2	Resultados das RNA com regularização de dados	19
4.3	Análise exploratória para o hiperparâmetro λ	25
5	Conclusões e Trabalhos Futuros	27
	REFERÊNCIAS	28
	APÊNDICE A	31

1 Introdução

Os fenômenos de transferência de calor e massa em uma torre de resfriamento de água por evaporação tornam este processo bastante desafiador do ponto de vista da modelagem em função da relação complexa entre variáveis como vazões, propriedades físicas e químicas e da dependência de parâmetros associados ao ar e à água. Tal complexidade também pode ser verificada em diversos outros equipamentos associados à indústria de processos, tais como reatores químicos, torres de destilação, evaporadores, etc. Ao mesmo tempo, o desenvolvimento de técnicas de ciência de dados com foco em inteligência artificial e Redes Neurais Artificiais (RNA) proporciona ferramentas que podem fornecer ótimos resultados para a análise destes processos diretamente a partir de dados históricos destes, sem a necessidade de aplicação das leis básicas que se adequem a cada situação estudada.

Nesse sentido, a aplicação de RNA na análise de processos industriais agrega maior facilidade e dinamismo aos estudos destes, permitindo a aquisição de resultados de forma simples. Assim, tendo em vista a aplicação relativamente recente destas ferramentas em processos industriais, verificou-se, em dados históricos associados ao funcionamento de uma torre de resfriamento, uma boa oportunidade de aplicação de RNA para realizar a predição da temperatura de saída da água na torre, explorando a aplicabilidade da ferramenta neste processo e verificando quais os parâmetros associados à rede que melhor se adequam a um modelo de predição em um processo industrial.

De forma a obter uma RNA com boa capacidade de generalização, é possível utilizar-se os métodos denominados técnicas de regularização durante a etapa de treinamento. Tais técnicas possibilitam uma modelagem mais parcimoniosa da rede, evitando que esta se adapte a características aleatórias dos dados e fornecendo a ela a possibilidade de atingir maiores capacidades de predição e generalização a partir dos dados de operação.

Este trabalho tem como estudo de caso uma torre industrial de resfriamento de água, para a qual, a partir de dados de entrada de operação tais como as vazões de ar e água, temperaturas de entrada na torre e a umidade do ar, será verificada a possibilidade de se prever a temperatura de saída da água na torre. Além disso, será estudado também o efeito de três técnicas diferentes de regularização sobre as redes (regularização L1, L2 e *dropout*) e dos valores dos hiperparâmetros associados às técnicas de regularização, de forma a obter a melhor capacidade de predição da variável de interesse. Desta forma, o objetivo do trabalho é a aplicação de redes neurais artificiais com regularização de dados de forma a obter um modelo preditivo eficiente para uma torre de resfriamento industrial visando à predição da temperatura de saída da água da mesma.

Para tanto, após as etapas iniciais de tratamento dos dados, são treinadas RNA's com diferentes arquiteturas, ainda sem a aplicação de técnicas de regularização de dados. Com isto, são definidas as melhores estruturas de rede em termos de métricas estatísticas que mensuram o desempenho da RNA. Posteriormente, é realizada a aplicação das técnicas de regularização de dados de forma a aumentar a capacidade de generalização das RNA selecionadas na etapa anterior. É obtido, deste modo, um modelo de rede neural com capacidade de predição satisfatória para o processo analisado.

2 Revisão Bibliográfica

Neste capítulo será realizada uma revisão de fundamentos sobre Inteligência Artificial, *Machine Learning*, *Deep Learning* e a técnica de Redes Neurais Artificiais, aplicando-os à predição de variáveis através de dados, juntamente com técnicas de regularização de dados para a melhoria da capacidade de generalização dos mesmos.

2.1 Fundamentos de *Machine Learning*, *Deep Learning* e Inteligência Artificial

2.1.1 *Inteligência Artificial*

A Inteligência Artificial (IA) é um ramo de estudos que tem como objetivo desenvolver ferramentas computacionais que executem atividades nas quais humanos são atualmente melhores, podendo, eventualmente, realizá-las com muito mais eficiência. Algumas destas atividades podem ser citadas como sendo: jogos digitais, robótica (por exemplo, carros autônomos), reconhecimento facial e de voz, entre outras aplicações (Venkatasubramanian, 2018).

De forma geral, um sistema de Inteligência Artificial deve, inicialmente, armazenar o conhecimento, ou seja, usualmente na forma de dados, que podem ser repassados de forma declarativa ou procedimental (um código ou equação que represente a informação denominada de conhecimento). Então, o sistema de IA deve ser capaz de resolver problemas através da utilização do aprendizado adquirido com o conhecimento armazenado e, por fim, ter a capacidade de adquirir novos conhecimentos através da experiência (Haykin, 2001).

2.1.2 *Machine Learning e Deep Learning*

Machine Learning (Aprendizado de Máquina) se refere a uma classe, dentro da Inteligência Artificial, de técnicas computacionais que permitem que computadores possam aprender através de experiências. Tais técnicas só são possíveis graças a um conjunto de pré-requisitos necessários para que o aprendizado possa ocorrer, como a existência de dados, geralmente vindos de simulações ou experimentos, preferencialmente em grandes quantidades, e de um modelo, o qual é um programa ou função matemática que descreve um sistema ou processo (Dobbelaere et. al, 2021).

Deep Learning (Aprendizagem Profunda), por outro lado, se refere a um conjunto de técnicas, dentro de *Machine Learning*, as quais utilizam Redes Neurais Artificiais (RNA) com um grande número de camadas internas da rede para treinamento e modelagem dos dados (Dobbelaere et. al, 2021). As RNA possibilitam o aprendizado de comportamentos e padrões entre os dados de entrada e saída inclusive quando estes são altamente não-lineares e complexos. Com isso, essa ferramenta permite boa flexibilidade e representatividade em relação aos resultados que podem ser alcançados (Himmelblau, 2000).

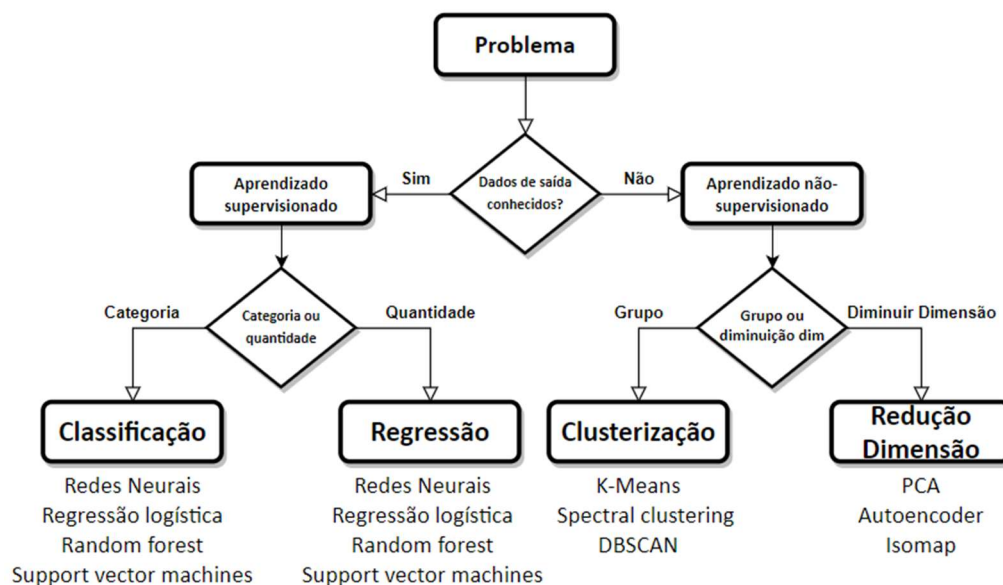
2.1.3 *Classificações de algoritmos em Machine Learning*

Existe uma ampla variedade de modelos e algoritmos, separados em diversos sub grupos, aplicados em projetos de *Machine Learning*. As formas de classificação dos tipos de algoritmo podem estar associadas ao propósito do modelo e o método de aprendizado deste. Em relação ao propósito, o modelo pode ser tanto de regressão, quando se deseja um valor

contínuo como o resultado, quanto de classificação, quando se deseja classificar os dados em dois ou mais grupos ou categorias (Dobbelaere et. al, 2021).

Em relação à metodologia utilizada para o aprendizado, têm-se dois tipos distintos, o primeiro chamado de aprendizado supervisionado e o segundo de aprendizado não-supervisionado. No caso dos modelos de aprendizado supervisionado, para cada dado de entrada são conhecidos os dados de saída, ou resposta, e o modelo deve, a partir desta informação, prever respostas para novos dados de entrada. Já no caso de modelos de aprendizado não supervisionado, o algoritmo objetiva reconhecer padrões a partir dos dados de entrada, sem possuir informações sobre os dados de saída (Dobbelaere et. al, 2021). A Figura 1 demonstra de forma visual a relação entre as técnicas e os algoritmos com os respectivos métodos de aprendizado.

Figura 1: Classificações de algoritmos em *Machine Learning*.



Fonte: Adaptado de

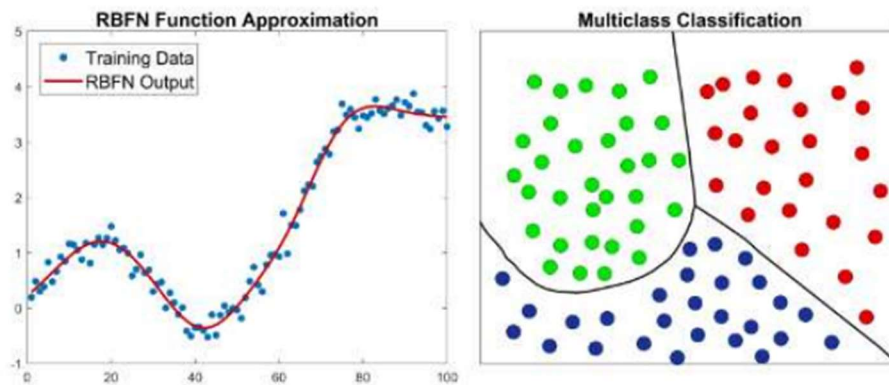
<https://pythonnumericalmethods.berkeley.edu/notebooks/chapter25.01-Concept-of-Machine-Learning.html> (Acesso em: 15/05/2022).

Em projetos de aprendizado não-supervisionado, um algoritmo bastante utilizado é o *K-Means*, o qual é uma técnica que busca realizar a clusterização, ou agrupamento, dos dados em dois ou mais grupos, dentro dos quais cada conjunto de dados possui características semelhantes entre si. Nesse sentido, é uma técnica de análise de dados explorativa, sendo capaz de encontrar semelhanças desconhecidas entre as estruturas dos dados (Govender et. al, 2019).

Os algoritmos de aprendizado supervisionado mais comumente utilizados são os modelos de regressão linear, as *support vector machines* (máquinas de vetores de suporte) e as Redes Neurais Artificiais. As Redes Neurais Artificiais possuem uma gama de aplicações bastante ampla, desde a possibilidade de realizar procedimentos de regressão e classificação até o processamento de imagens (Dobbelaere et. al, 2021). A Figura 2 representa graficamente

exemplos de resultados de aplicação de modelos de regressão e de classificação aplicados a dados.

Figura 2: Representação gráfica dos resultados de técnicas de regressão (esquerda) e de classificação (direita).



Fonte: Adaptado de Venkatasubramanian, 2018.

2.2 Redes Neurais Artificiais (RNA)

2.2.1 Fundamentos sobre Redes Neurais Artificiais

O cérebro humano possui uma capacidade de processamento de dados bastante complexa e não-linear, diferenciando-se, em muitos aspectos, da forma de processamento de informações de um computador convencional. Esta capacidade de processamento do cérebro humano está associada à forma de ação de seus componentes estruturais principais, os neurônios, formados pelo corpo celular, dendritos e axônios. Os neurônios possuem uma capacidade de realizar certos tipos de processamentos de informação, como o reconhecimento de padrões, de forma mais rápida que os computadores atuais (Haykin, 2001).

Inspiradas pela forma de funcionamento do cérebro humano, as RNA podem ser consideradas como modelos empíricos não-lineares capazes de determinar padrões entre dados, ou realizar outras formas de processamento de informação como regressão e classificação, com dados de entrada (*input*) e dados de saída (*output*) (Himmelblau, 2000). Com isso, as RNA são aplicadas como modelos denominados de *Black-Box Models*, ou seja, modelos nos quais o funcionamento interno não tem semelhança física com o sistema que está sendo analisado (Dobbelaere et. al, 2021).

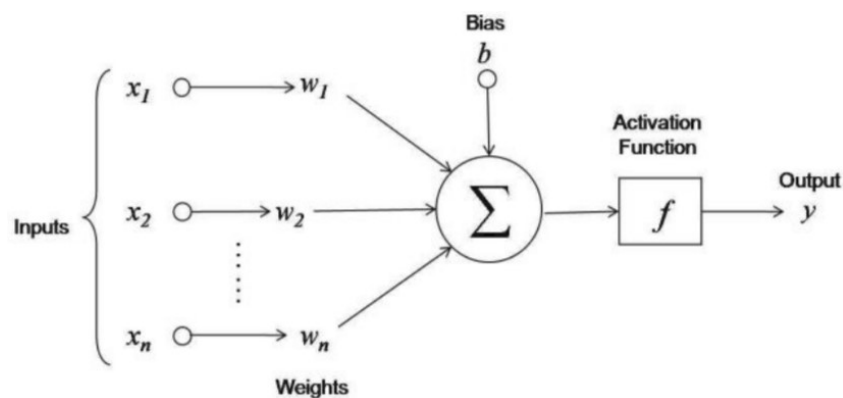
2.2.2 Estruturas de Redes Neurais Artificiais

A unidade de processamento de uma RNA é o neurônio, denominado no sistema de uma rede como nó, o qual recebe dados de entrada (*input*) e retorna dados de saída (*output*). Os dados de entrada, ou sinais de entrada, podem ser originados de outros nós ou de outras fontes de dados no sistema. Cada entrada é ponderada por valores denominados pesos (*weights*) e, então, os sinais de entrada, multiplicados pelos seus respectivos pesos, são somados. Nesta etapa também ocorre a soma, ao sinal ponderado, de um parâmetro externo

denominado *bias*, o qual tem a função de aumentar ou diminuir a magnitude do sinal (Himmelblau, 2000).

Com isso, o sinal é submetido a uma função escalar denominada de função de ativação (*activation function*), que restringe o intervalo dos dados, ou sinais, de saída, caracterizando, enfim, a obtenção do *output* de acordo com os dados de entrada. A função de ativação pode ser de diferentes tipos, como uma função normalizadora a um intervalo de $[0, 1]$ ou até uma função linear (Haykin, 2001). Com a Figura 3, é possível visualizar a estrutura de um nó existente em uma RNA, relacionando o fluxo dos dados com cada etapa de seu processamento.

Figura 3: Estruturas de um único neurônio (nó) em uma Rede Neural Artificial



Fonte: <https://www.datacamp.com/tutorial/neural-network-models-r>
(Acesso em: 23/05/2022).

Assim, de forma geral, a relação entre os sinais de entrada x_i de um neurônio e o sinal de saída, y , é dada por:

$$y = \varphi(b + \sum_i^m w_i \cdot x_i) \quad (2.1)$$

Onde:

y = sinal de saída (*output*);

φ = função de ativação;

b = bias;

x_i = sinais de entrada (*input*);

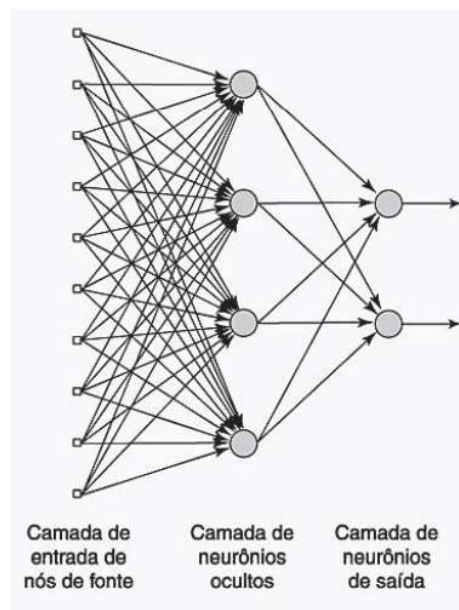
w_i = pesos aplicados aos sinais de entrada.

As Redes Neurais Artificiais são compostas por diversas destas unidades de processamento, ou nós, os quais são dispostos em camadas. Os nós localizados na entrada da rede, que recebem dados de uma fonte externa, estão em uma camada denominada camada de entrada (*input layer*), e os nós localizados na saída da rede em uma camada denominada

camada de saída (*output layer*) (Zhang et. al, 1998). Ainda é possível a existência de nós em camadas intermediárias na rede, denominados como nós ocultos (*hidden layers*), de forma que as camadas nas quais estes nós estão localizados são denominadas camadas ocultas (*hidden layers*) (Himmelblau, 2000).

Nesse sentido, a arquitetura de Redes Neurais Artificiais pode ser classificada, principalmente, em relação à quantidade de camadas e a forma de conexão entre estas. Assim, pode-se definir, em geral, três tipos de arquitetura para as redes: redes *feedforward* de camada única, redes *feedforward* de múltiplas camadas, cujo exemplo de estrutura pode ser visualizado na Figura 4, e redes recorrentes. Nas redes do tipo *feedforward*, o sinal é sequencialmente ponderado e repassado para a camada seguinte, ou seja, cada nó tem como sinal de entrada somente a soma ponderada dos sinais de saída da camada anterior. Já nas redes recorrentes existe, pelo menos, um laço (ou conexão) de realimentação, operação a qual possui um grande impacto nas características da rede (Himmelblau, 2000). Ainda, nas redes recorrentes, estão presentes elementos de atraso unitários em certas conexões, os quais permitem um comportamento dinâmico não-linear (Haykin, 2001).

Figura 4: Rede Neural Artificial tipo *feedforward* de múltiplas camadas constituída de dez neurônios de entrada, quatro neurônios ocultos e dois neurônios de saída.



Fonte: Haykin, 2001.

O conhecimento do número de parâmetros, em uma RNA, que devem ser determinados durante o processo de aprendizagem auxilia o entendimento de o quão complexa é a rede. Assim, o número de parâmetros a serem determinados, que estão relacionados aos pesos (*weights*) em cada conexão juntamente com os valores do *bias* em cada neurônio, podem ser calculados de forma simples com o conhecimento do número de neurônios em cada camada da rede. Em cada camada, pode-se determinar o número de parâmetros através da multiplicação entre a quantidade de dados de saída e a quantidade de dados de entrada acrescidos de uma unidade (referente ao *bias*) (Haykin, 2001).

A quantidade de neurônios de entrada na rede é igual ao número de variáveis de entrada no sistema a ser modelado, enquanto a quantidade de neurônios de saída é igual ao número de variáveis que se deseja prever. Já para a quantidade de neurônios e camadas ocultas, o processo de escolha é mais complexo. Se for utilizado um número de neurônios ocultos muito pequeno, a rede pode apresentar baixa capacidade de processamento e aprendizado dos dados. Por outro lado, se for utilizado um número muito elevado de neurônios ocultos, a rede pode ter uma capacidade excessiva de processamento, proporcionando um aprendizado de aspectos irrelevantes dos dados, prejudicando a generalização e a predição final.

Nesse sentido, o ideal é utilizar o menor número possível de neurônios ocultos que possam realizar o processo de aprendizagem da forma adequada. Uma diretriz possível de ser seguida, pelo menos preliminarmente, é a regra da pirâmide geométrica, a qual diz que a quantidade de neurônios segue a forma de uma pirâmide, ou seja, com o número neurônios na rede diminuindo desde a entrada até a saída.

Esta diretriz, em casos que o problema a ser resolvido é complexo e ao mesmo tempo não há uma quantidade grande de neurônios de entrada e saída, pode subestimar o número de neurônios ocultos. Assim, o processo de definição do número de neurônios ocultos na rede pode utilizar essa informação como uma quantidade inicial, a qual pode ser incrementada durante experimentos. Assim, sendo o número de neurônios de entrada denotado por n e o número de neurônios de saída denotado por m , em estruturas com apenas uma camada oculta, esta camada terá um número de neurônios ocultos igual a $\sqrt{m \cdot n}$. Agora, se a estrutura da rede possuir duas camadas ocultas, a definição segue a regra abaixo (Masters, 1993):

$$r = \sqrt[3]{n/m} \quad (2.2)$$

$$NHID_1 = m \cdot r^2$$

$$NHID_2 = m \cdot r$$

Onde:

n = número de neurônios de entrada na rede;

m = número de neurônios de saída na rede;

r = parâmetro definido a partir da razão entre o número de neurônios de entrada e saída;

$NHID_1$ = número de neurônios na primeira camada oculta;

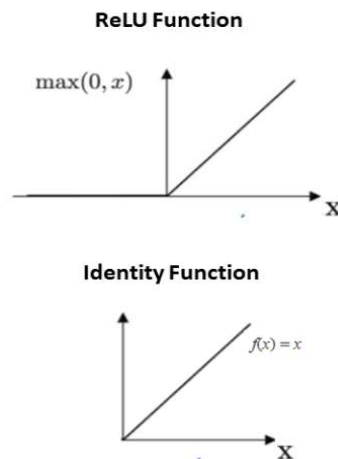
$NHID_2$ = número de neurônios na segunda camada oculta.

2.2.3 O processo de aprendizado nas Redes Neurais Artificiais

Em estudos de regressão com Redes Neurais Artificiais, quando são utilizadas redes com múltiplas camadas do tipo *feedforward*, geralmente é utilizada nas camadas ocultas a função de ativação ReLU (*Rectified Linear Unit*), a qual retorna 0 para valores de entrada negativos, devido à boa performance dos resultados e baixo custo computacional (Goodfellow et. al, 2016). Ao mesmo tempo, na camada de saída geralmente é utilizado a função linear como a função de ativação, visto que esta pode retornar um amplo intervalo de sinais de saída (Zhang

et. al, 1998). Para fins de comparação, ambas as funções de ativação estão representadas graficamente na Figura 5.

Figura 5: Representação das funções de ativação ReLU e Linear



Fonte: Adaptado de <https://www.datacamp.com/tutorial/neural-network-models-r> (Acesso em 30/05/2022).

No processo de aprendizado supervisionado de uma rede, o conjunto de dados é separado em três conjuntos distintos: de treinamento, de validação e de teste. O conjunto de dados de treinamento é utilizado no processo de definir, através de certos procedimentos, os valores dos pesos e do *bias* em cada neurônio de forma a garantir a melhor representação do comportamento dos dados de entrada e saída (Himmelblau, 2000). O conjunto de dados de validação tem a função de, através de métricas estatísticas, medir o desempenho do modelo treinado durante o processo de aprendizado da rede, comparando os dados de saída preditos com os reais. Por fim, o conjunto de dados de teste avalia o desempenho da rede após o seu treinamento completo com o objetivo de verificar a capacidade de predição e generalização da rede com dados não empregados anteriormente (Milléo, 2020).

No processo de aprendizado da rede, os dados são propagados através dos neurônios, sendo ponderados pelos pesos, somados ao *bias* e submetidos à função de ativação em cada camada, e então, na saída da rede, ocorre a comparação destes valores de saída com os dados reais associados aos dados de entrada processados (Alves, 2003). Essa comparação entre os dados é feita através de uma função denominada função de perda (*loss function*). Nesse sentido, o processo de aprendizado pode ser considerado um procedimento iterativo de ajuste dos pesos e dos valores de *bias* de forma a minimizar a função de perda, garantindo uma melhor predição dos dados de saída. Diversas funções estatísticas podem ser utilizadas como a função de perda para análise do desempenho e aprendizado de uma RNA. Entre elas podem ser citadas, principalmente, o erro médio quadrático (MSE), a raiz quadrada do erro médio quadrático (RMSE) e o erro absoluto médio (MAE) (Siqueira, 2021).

Para a definição do processo de aprendizado da rede também é importante a escolha dos valores de dois hiperparâmetros, o número de iterações (*epochs*) e o *batch size*. O número de *epochs* está associado ao número de vezes que o algoritmo de aprendizado da rede irá

percorrer os dados do conjunto de treinamento e calcular o valor da função de perda. Já o *batch size* está associado ao número de amostras (instâncias ou linhas de dados) que são submetidos à rede a cada iteração, ou seja, antes de realizar o ajuste dos parâmetros internos do modelo (Lee & Song, 2019).

2.2.4 Trabalhos com aplicação de Redes Neurais Artificiais em modelos de processo

Diversos trabalhos e estudos foram realizados utilizando Redes Neurais Artificiais para a predição (regressão ou classificação) de variáveis de interesse em processos industriais de diferentes tipos. Entre estas aplicações, podem ser citados estudos relacionados à modelagem de um processo em uma coluna de destilação binária (Singh et. al, 2017), à predição da formação de azeótropos em misturas binárias (Alves, 2003) e à modelagem de reatores multifásicos (Tarca et. al, 2003).

Singh et. al (2017) analisaram a aplicação de redes neurais recorrentes com duas camadas para modelar o funcionamento de uma coluna de destilação binária em modo de operação contínuo. Nesse sentido, a RNA construída fornecia a previsão da composição do destilado com grande acurácia, representando um modelo que fornece boas previsões apesar das diferentes não-linearidades que ocorrem em processos de destilação.

Alves (2003) buscou aplicar redes do tipo *feedforward* com três camadas de forma a realizar a predição da formação de azeótropos em sistemas binários. Assim, concluiu-se que a técnica de RNA consegue desempenhar bem a função de predizer a formação (ou não) de azeótropos a partir de dados adequados dos componentes puros sem conhecimentos sobre a não-idealidade das misturas.

Tarca et. al (2003) estudaram a aplicação de RNA na modelagem de reatores químicos multifásicos de leito recheado com escoamento em contracorrente de forma a predizer variáveis como a queda de pressão no reator, coeficiente de transferência de massa e outros parâmetros. Foram utilizados para comparação resultados fenomenologicamente consistentes (extensas bases de dados experimentais obtidos na literatura), ou seja, os resultados que seguem leis físicas que descrevem o sistema. Assim, diversos modelos de Redes Neurais Artificiais (com diferentes configurações de dados de entrada) conseguiram atingir bons resultados, ajustando-se corretamente aos dados fisicamente conhecidos.

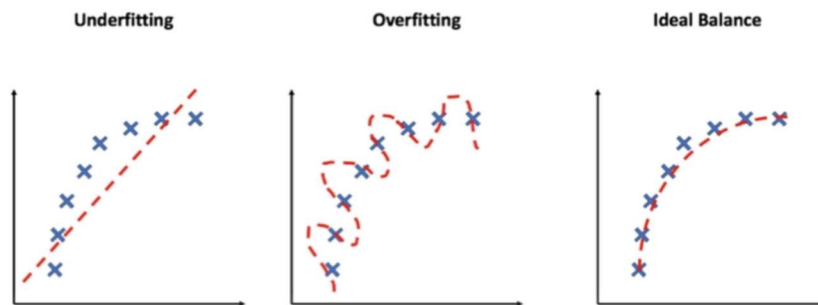
2.3 Regularização de Dados

2.3.1 Sobre-ajuste e Sub-ajuste

O objetivo da rede no problema de regressão, ao ser treinada, é aprender os padrões do processo que relaciona os dados de entrada com os de saída de forma a realizar generalizações para dados de entrada não usados no treinamento. Nesse sentido, considerando que dados de treinamento em geral contém ruídos e distúrbios, a RNA não deve reproduzir as características aleatórias dos dados utilizados para seu treinamento, pois isso prejudicaria a capacidade de generalização (Nusrat & Jang, 2018). Neste caso, temos o fenômeno de sobre-ajuste (*overfitting*), o qual pode ocorrer em situações em que os dados de treinamento não representam bem o conjunto dos possíveis dados de entrada, ou quando o modelo preditivo é complexo demais para os dados de treinamento empregados. Por outro lado, quando estão disponíveis poucos dados de treinamento, impedindo o modelo de aprender efetivamente,

resultando em um aprendizado incompleto, ocorre o fenômeno de sub-ajuste (*underfitting*) (Phaisangittisagul, 2016). A comparação gráfica entre o ajuste ótimo de um modelo a um conjunto de dados e os possíveis sobre-ajuste e sub-ajuste deste modelo são demonstrados na Figura 6.

Figura 6: Representação de sub-ajuste, sobre-ajuste e um ajuste ideal a dados, respectivamente.



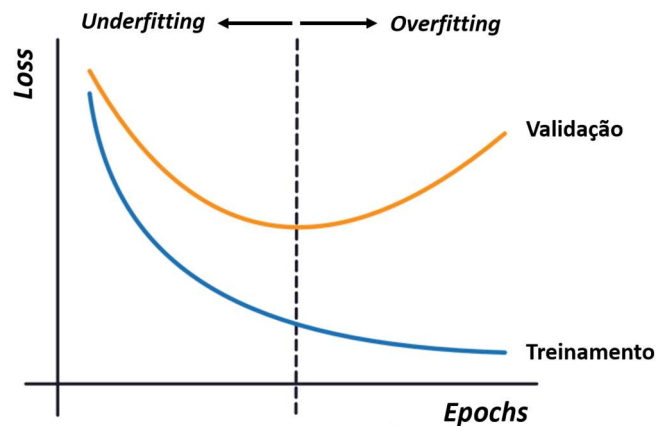
Fonte: <https://medium.com/geekculture/investigating-underfitting-and-overfitting-70382835e45c> (Acesso em 07/06/2022).

2.3.2 Curvas de aprendizado

No aprendizado de máquina, a curva de aprendizado demonstra a performance de aprendizado e generalização do modelo que está sendo treinado em função do número de iterações. Nesse sentido, para as Redes Neurais Artificiais, geralmente a curva de aprendizado demonstra o comportamento da função de perda (*loss function*) em função do número de iterações de treinamento. A curva de aprendizado deve comparar a performance de aprendizado do modelo em relação aos dados de treinamento e a dados não vistos antes (dados de validação), medindo a capacidade de generalização da rede (Perlich, 2009; Domhan et. al, 2015).

Um modelo de rede deve, ao longo das *epochs*, apresentar uma diminuição do valor da função de perda, tanto para os dados de treinamento quanto para os dados de validação. Contudo, quando ocorrem casos de sobre-ajuste (*overfitting*) dos dados, o valor da função de perda para os dados de validação começa, a partir de um certo ponto, a aumentar ou se manter estável, mas consideravelmente maior que a curva dos dados de treinamento, enquanto o valor da função de perda para os dados de treinamento se mantém estável (ou diminuindo ainda mais), como pode ser visualizado na Figura 7.

Figura 7: Representação qualitativa de uma curva de aprendizado com dados de treinamento e de validação.



Fonte: Adaptado de <https://www.kaggle.com/code/ryanholbrook/overfitting-and-underfitting/tutorial> (Acesso em 11/06/2022).

2.3.3 Técnicas de Regularização de Dados

A regularização de dados atua na diminuição do efeito negativo do sobre-ajuste sobre a capacidade de generalização do modelo treinado. Nesse sentido, as técnicas de regularização de dados promovem uma “suavização” da função do modelo obtido, de forma a garantir uma capacidade de generalização significativa. Em outras palavras, a regularização de dados proporciona penalizações aos parâmetros da rede que atribuem uma complexidade excessiva ao modelo, atuando, por exemplo, sobre os pesos associados aos nós da rede.

Matematicamente, o termo de penalização aparece somado ao termo que representa a função de perda (*loss function*), resultando na função de perda regularizada, a qual deve ser minimizada de forma a encontrar, com o processo de aprendizado, a melhor solução que busca uma generalização efetiva dos dados. Os principais modos de regularização de dados que utilizam essa estrutura são denominados regularização L1 (representado na equação 2.3), também chamada de Lasso (originado da sigla em inglês *least absolute shrinkage and selection operator*), e regularização L2 (representado na equação 2.4), também chamada regressão Ridge (Demir-Kavuk et. al, 2011).

$$J_1 = L(y, \bar{y}) + \lambda \sum_{i=1}^N |w_i| \quad (2.3)$$

$$J_2 = L(y, \bar{y}) + \lambda \sum_{i=1}^N w_i^2 \quad (2.4)$$

Onde:

J = função a ser minimizada;

L = função de perda (*loss function*);

y e \bar{y} = dado de saída real e dado de saída predito, respectivamente;

λ = parâmetro de regularização;

w = pesos (*weights*) em cada neurônio.

No caso da técnica de regularização L1, pode-se perceber, pela equação 2.3, que o termo associado à regularização leva em conta o somatório dos valores absolutos dos pesos na função objetivo, enquanto a técnica de regularização L2, pela análise da equação 2.4, considera no cálculo o somatório dos quadrados dos pesos. A técnica L1 possibilita a redução de alguns parâmetros de peso a 0, removendo a interferência deste parâmetro do modelo, ou seja, eliminando esta conexão da rede (Demir-Kavuk et. al, 2011). A técnica L2, por outro lado, força valores de peso a serem mínimos, contudo, não proporciona a conversão destes valores de peso a 0, como na técnica L1 (Boyco & Moroz, 2021).

Outra técnica mais recente de regularização de dados em RNA é a técnica denominada *dropout*. Esta técnica previne a ocorrência do sobre-ajuste (*overfitting*) através da remoção da interferência de parâmetros de peso (convertendo-os a 0) aleatoriamente durante o treinamento da RNA (Badola et. al, 2020). Nesta técnica, a probabilidade de manutenção de um valor de peso na rede pode ser controlada através do hiperparâmetro p (definido pelo usuário antes do processo de treinamento da rede), o qual pode variar na faixa de 0 a 1. Nesse sentido, elevados valores de p (mais próximos de 1) representam um menor efeito de *dropout*, enquanto valores menores de p (mais próximos de 0) representam um maior efeito de *dropout* (Srivastava et. al, 2014).

2.3.4 Trabalhos de aplicação de Rede Neurais Artificiais com Regularização de Dados

Podem ser encontrados na literatura trabalhos os quais, através do uso de Redes Neurais Artificiais juntamente com técnicas de regularização de dados, buscam a predição de dados utilizando regressões em diversos sistemas. Entre tais estudos podem ser citados sistemas como a previsão da carga de refrigerante líquido em bombas de calor (Hwang et. al, 2020) e a previsão da quantidade de transferência de calor para metano em câmaras de resfriamento regenerativo (Waxenegger-Wilfing et. al, 2020).

Hwang et. al (2020) analisaram a predição da carga de líquido refrigerante utilizada em bombas de calor, a qual, quando em quantidade incorreta, pode ocasionar falhas. Nesse sentido, foi utilizada a técnica de RNA, juntamente com as técnicas de regularização L1 e L2, de forma que, ao utilizar uma rede com a aplicação destas técnicas de regularização de dados, foi possível alcançar uma performance e capacidade de generalização maiores em relação à rede original.

Waxenegger-Wilfing et. al (2020) estudaram o treinamento de uma RNA para realizar a predição da temperatura da parede no interior da câmara de combustão de um foguete espacial. Neste estudo, também foi analisado a aplicação da técnica de regularização L2 para verificar os impactos na capacidade de generalização do modelo da rede. Com isso, foi incluído no modelo final a aplicação da técnica L2, o que garantiu uma rede eficiente e que conseguia atuar com sucesso em dados não apresentados anteriormente.

3 Materiais e Métodos

Neste capítulo, serão apresentadas as ferramentas computacionais (*softwares*), a base de dados e os métodos de análise utilizados no desenvolvimento do trabalho.

3.1 Softwares

Para o desenvolvimento dos modelos de inteligência artificial (Redes Neurais Artificiais) foi utilizada a linguagem de programação Python (v 3.9.7) através da IDE (*Integrated Development Environment*) Jupyter Notebook (v 6.4.5). Para o tratamento dos dados e construção das RNA foram importadas as seguintes bibliotecas: Pandas, NumPy, Tensorflow, Scikit-learn e Matplotlib. Todo desenvolvimento foi realizado em um computador com processador Intel Core i5 2.5 GHz, memória RAM de 8 GB e sistema operacional Windows 10. Os códigos desenvolvidos podem ser encontrados no Apêndice A.

3.2 Base de Dados

No trabalho, foram utilizados dados históricos do funcionamento de uma torre de resfriamento do tipo tiragem induzida de uma indústria petroquímica localizada no estado do Rio Grande do Sul, a qual funciona com a atuação de 12 ventiladores que auxiliam no controle da vazão de ar na torre. A água resfriada que deixa a torre abastece áreas da indústria como o sistema de resfriamento de uma central de utilidades, uma planta de craqueamento de nafta e uma planta de purificação de aromáticos. Em relação a parâmetros operacionais da torre de resfriamento, podem ser citadas a vazão máxima de água, igual a 35.000 m³/h, a temperatura máxima da água quente que chega à torre, igual a 45 °C e os limites inferior e superior de temperatura da água que deixa a torre, iguais a 28 °C e 30 °C, respectivamente. Os dados foram adquiridos no período de junho de 2014 a maio de 2015, resultando em um número de amostras na base de dados utilizada igual a 1672.

As variáveis que constam na base de dados utilizada são as vazões de água e de ar que entram na torre de resfriamento, as temperaturas de entrada e saída da água, a umidade do ar atmosférico e as temperaturas de bulbo seco e úmido do ar que entra na torre. Com isso, o conjunto de dados utilizado possui um total de 7 variáveis, sendo a temperatura de saída da água considerada o *output*, enquanto o restante dos atributos (6) são os *inputs* do sistema estudado. A Tabela 1 fornece maiores detalhes em relação às variáveis analisadas no estudo.

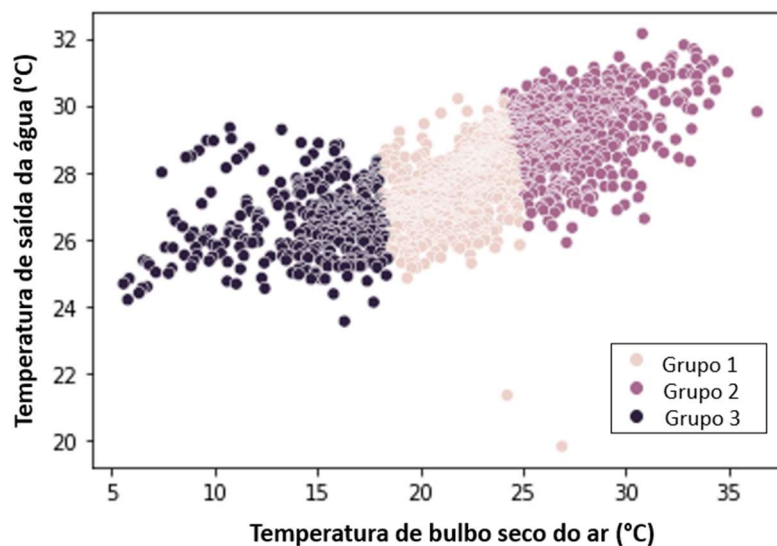
Tabela 1: Detalhamento das variáveis analisadas no processo.

Variável	Unidade	Símbolo	Tipo
Vazão de entrada da água	kg/s	Q _{água}	<i>input</i>
Vazão de entrada do ar	kg/s	Q _{ar}	<i>input</i>
Umidade relativa do ar	%	φ	<i>input</i>
Temperatura de entrada da água	°C	T _{in_água}	<i>input</i>
Temperatura de bulbo seco do ar na entrada	°C	T _{bs_ar}	<i>input</i>
Temperatura de bulbo úmido do ar na entrada	°C	T _{bu_ar}	<i>input</i>
Temperatura de saída da água	°C	T _{out_água}	<i>output</i>

3.3 Procedimento de definição dos dados de treinamento, teste e validação

Visando o teste posterior das técnicas de regularização de dados, foi feita a separação da base de dados (clusterização) em três grupos distintos (Grupo 1, Grupo 2 e Grupo 3) com base no algoritmo K-Means através da biblioteca Scikit-learn. Para a separação nos grupos, foram utilizadas as variáveis temperatura de bulbo seco do ar ($^{\circ}\text{C}$) e temperatura de saída da água ($^{\circ}\text{C}$). A escolha destas variáveis se deve ao fato de que a primeira se refere às condições de entrada do processo enquanto a segunda caracteriza o *output* do processo. Esta escolha possibilita uma visualização e separação intuitivas dos pontos de operação na torre de resfriamento. Além disso, o resultado do agrupamento utilizando estas variáveis, que pode ser visto na Figura 8, resultou em uma dispersão dos dados bastante evidente, caracterizando grupos com boa separação dentre si. Após a divisão da base de dados nestes grupos, foi realizada a normalização dos dados de entrada.

Figura 8: Resultado da clusterização do conjunto de dados original, com base no algoritmo *K-Means*, utilizando a temperatura de saída da água ($^{\circ}\text{C}$) e a temperatura de bulbo seco do ar ($^{\circ}\text{C}$).



Fonte: Elaborado pelo autor.

De forma a aumentar as chances de ocorrência de sobre-ajuste e ao mesmo tempo simular uma situação na qual se possui apenas dados históricos em uma certa região de operação da torre de resfriamento, os grupos obtidos com o algoritmo *K-Means* foram associados aos conjuntos de dados de treinamento, teste e validação de modo que os dados de treinamento estivessem o mais distantes possível dos dados de teste. Isto faz com que seja necessária uma capacidade de generalização mais elevada da rede. Assim, o Grupo 1 foi associado ao conjunto de validação, o Grupo 2 foi associado ao conjunto de teste e o Grupo 3 foi associado ao conjunto de treinamento.

3.4 Definição da estrutura da RNA e seu treinamento

A construção da arquitetura da RNA foi realizada utilizando a função *Sequential* da biblioteca Tensorflow (Keras), que permite definir os hiperparâmetros de cada camada da rede individualmente, tais como o número de nós e a função de ativação, por exemplo. Em

todas as camadas ocultas foi utilizada a função de ativação ReLU, enquanto na camada de saída foi usada a função de ativação linear. Foram testados três números diferentes de nós ocultos com 1 camada oculta (HL) e cinco números de nós ocultos com 2 camadas ocultas, conforme a regra proposta por Masters (1993). Para as redes com 1 camada oculta, foram testadas arquiteturas com 3, 4 e 5 nós ocultos, enquanto para as redes com 2 camadas ocultas foram testadas redes com 3-2, 4-3, 5-4, 10-8 e 20-16 nós ocultos. Em todos os casos, de forma a possibilitar a comparação entre os resultados de cada arquitetura, foi utilizado um valor de número de *epochs* igual a 2500 e um valor de *batch_size* igual a 64.

Para a função de perda, foi utilizada a função de erro absoluto médio (*Mean Absolute Error* – MAE), que calcula a diferença absoluta média entre os valores de saída preditos e os valores reais, definido na Equação 3.1. Foi utilizada a função da raiz quadrada do erro médio (*Root Mean Square Error* – RMSE, Equação 3.2) como uma métrica que auxilia, através do fornecimento da informação do quanto os resultados obtidos estão esparsos entre si, na avaliação do desempenho da rede ao longo do processo de treinamento. Adicionalmente, esta informação auxilia na interpretação da curva de aprendizado e na qualidade de predição da rede desenvolvida. A raiz quadrada do erro médio representa a distância média entre os pontos de dados preditos o conjunto de dados reais.

$$MAE = \frac{\sum_{i=1}^N |\bar{y}_i - y_i|}{N} \quad (3.1)$$

$$RMSE = \sqrt{\sum_{i=1}^N \frac{\sum_{i=1}^N (\bar{y}_i - y_i)^2}{N}} \quad (3.2)$$

Onde:

N = número total de conjuntos de dados;

\bar{y}_i = valor de saída predito;

y_i = valor de saída real.

Mais precisamente, o MAE fornece a informação de o quanto os resultados preditos variam, de forma absoluta, dos valores reais presentes no conjunto de teste, enquanto o RMSE fornece a informação da presença de valores muito esparsos, também denominados *outliers*, representando o quanto os erros das predições variam entre si.

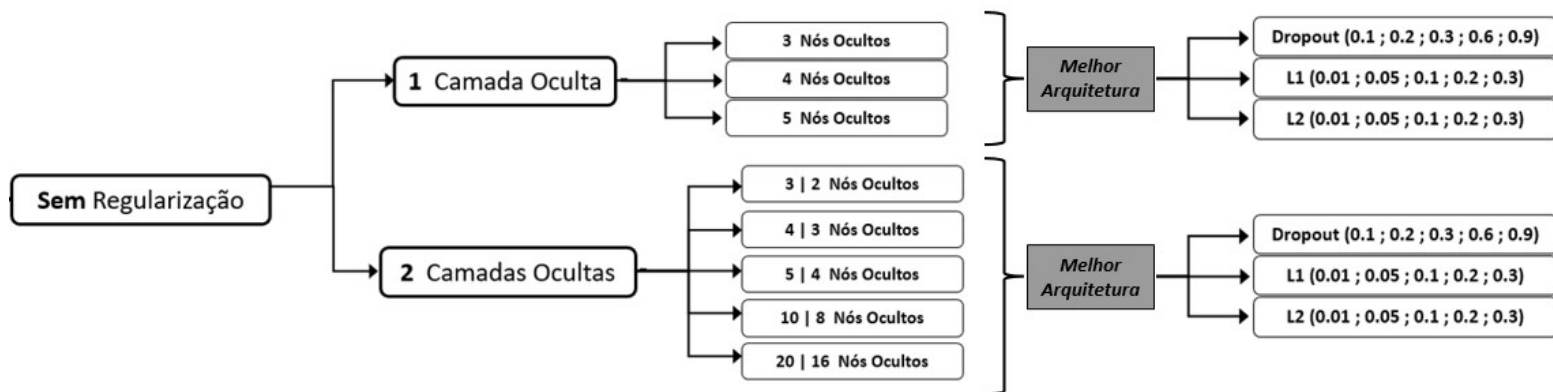
Com isso, ao final de cada treinamento das redes, foram obtidas as curvas de aprendizado, utilizando tanto o MAE quanto o RMSE. O resultado em termos de MAE e RMSE da aplicação dos dados de teste à rede desenvolvida foi utilizado para posterior comparação entre as arquiteturas utilizadas.

3.4.1 Aplicação de técnicas de regularização de dados

A partir dos resultados obtidos na etapa anterior com as RNA, foram aplicadas técnicas de regularização na rede que teve o melhor desempenho, ou seja, os menores valores de MAE e RMSE, tanto no caso com uma quanto com duas camadas ocultas, de forma a visualizar o impacto da regularização de dados na capacidade de generalização das arquiteturas. Com isto, diminui-se o grau de sobre-ajuste que ocorre nas redes com os melhores desempenhos de predição, obtendo, ao fim, as RNA mais eficientes.

Para as técnicas L1 e L2 de regularização, foram testados valores de λ descritos na literatura (Demir-Kavuk et. al, 2011; Badola et. al, 2020) de 0,01; 0,05; 0,1; 0,2 e 0,3. De forma semelhante, para a técnica *dropout* de regularização, foram testados valores de p descritos na literatura (Badola et. al, 2020) de 0,1; 0,2; 0,3; 0,6 e 0,9. Após a obtenção dos dados, foi feita uma análise exploratória dos valores de hiperparâmetro (λ ou p) que proporcionaram resultados de predição melhores que no caso sem regularização para aquela arquitetura de rede. A Figura 9 apresenta um resumo gráfico das etapas dos experimentos realizados.

Figura 9: Etapas de experimento e análise realizadas no estudo.



Fonte: Elaborado pelo autor.

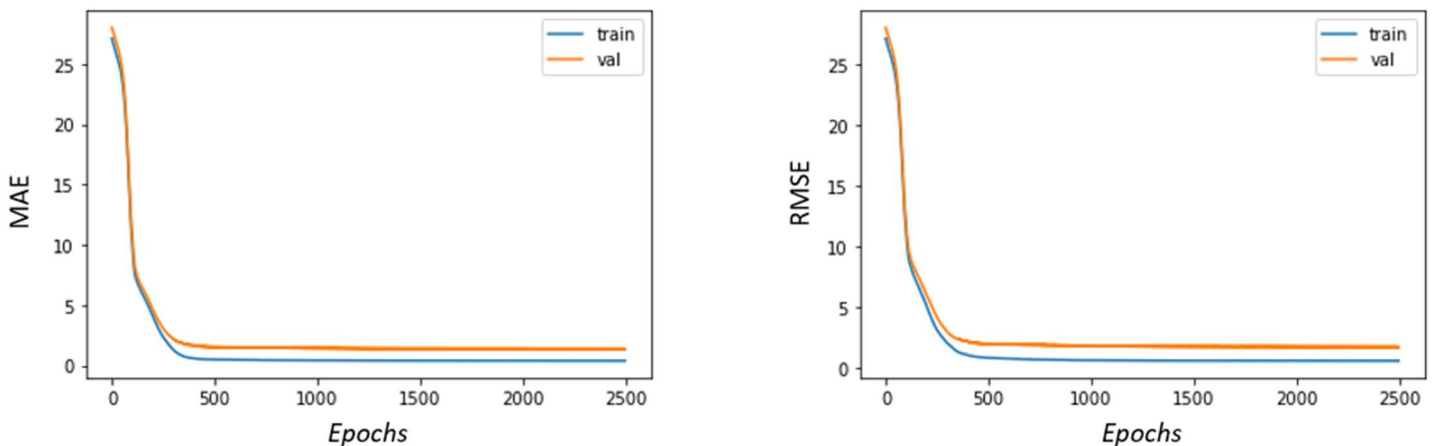
4 Resultados

Neste capítulo, serão apresentados os resultados obtidos para a performance das RNA construídas, treinadas e testadas. As redes foram desenvolvidas com o uso de conjuntos de treinamento e teste que proporcionassem o aumento das chances de ocorrer sobre-ajuste dos dados, de forma a testar as técnicas de regularização. O objetivo deste procedimento é simular a situação na qual o usuário dispõe apenas de dados históricos em uma dada região de operação da torre para o desenvolvimento do modelo.

4.1 Resultados das RNA sem regularização de dados

Tanto as redes com uma quanto com duas camadas ocultas resultaram, após o processo de treinamento, em curvas de aprendizado em termos de MAE/RMSE com comportamentos semelhantes, nas quais foi possível observar, mesmo que em pequena intensidade, a ocorrência de *overfitting*. Apesar de que, com o aumento do número de épocas, a curva associada à função de perda do conjunto de validação não tenha apresentado crescimento nos casos testados, é possível perceber que esta curva se manteve constante e maior que a curva do conjunto de treinamento. Nesse sentido, é possível verificar um sobre-ajuste em pequena escala. A Figura 10 demonstra exemplos das curvas de aprendizado obtidas em uma das configurações de rede analisadas.

Figura 10: Curvas de aprendizado obtidas no treinamento da RNA com duas camadas ocultas compostas por 4 e 3 nós. À esquerda, a curva associada à função perda (MAE) e à direita a curva associada à métrica RMSE.



Fonte: Elaborado pelo autor.

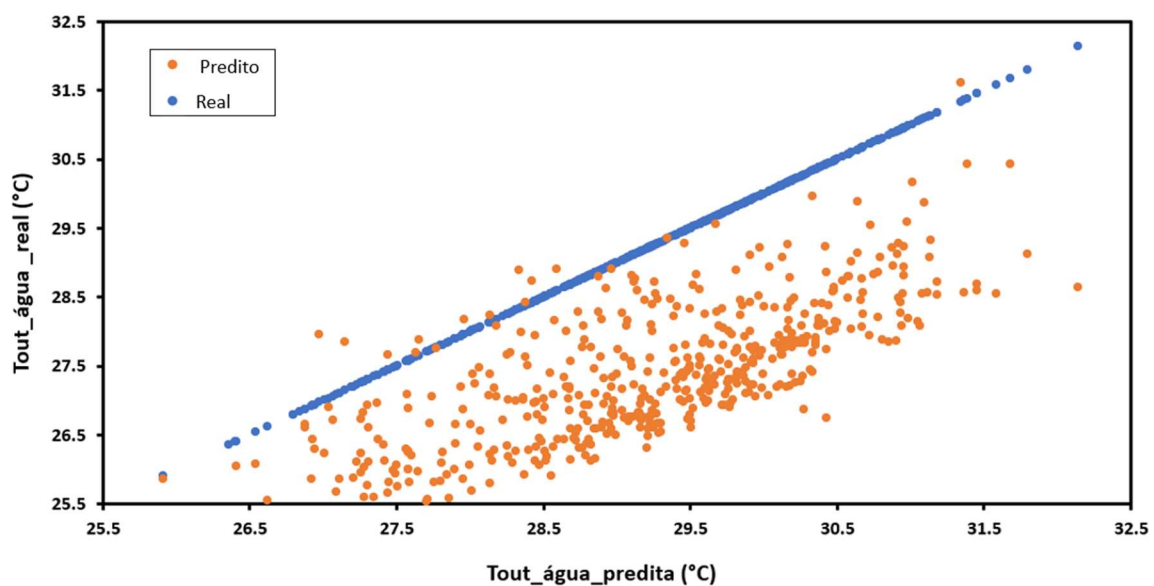
É possível verificar, na Tabela 2, que no caso das redes com uma camada oculta, a rede com 5 nós internos apresentou os melhores resultados de predição, enquanto no caso das redes com duas camadas ocultas, a rede com 5 e 4 nós apresentou os melhores resultados. As redes que apresentaram os piores desempenhos em cada caso, com uma ou duas camadas ocultas, foram as RNA com 4 nós e 10 e 8 nós ocultos, respectivamente.

Tabela 2: Resultados da aplicação das RNA treinadas aos dados de teste

Nº HL	Nº de Nós	MAE	RMSE
1	3	1,87	2,11
1	4	9,84	10,39
1	5	1,83	1,98
2	3 2	1,84	2,02
2	4 3	1,69	1,95
2	5 4	1,49	1,65
2	10 8	2,36	2,87
2	20 16	1,57	1,87

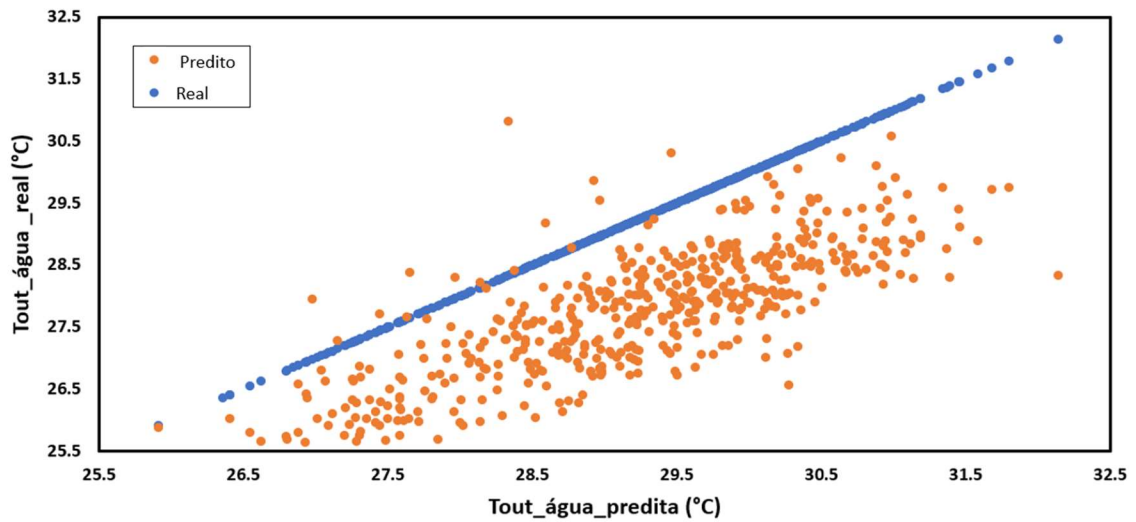
Para aprofundar a análise das redes que obtiveram os melhores resultados, as Figuras 11 e 12 mostram as diferenças entre os dados reais e os dados preditos pelas RNA. Os valores de MAE obtidos para as redes com 5 nós ocultos (MAE = 1,83) e com 5 e 4 nós ocultos (MAE = 1,49) permitem verificar que as diferenças entre as temperaturas de saída da água reais e preditas são menores na rede com duas camadas. De fato, isso pode ser verificado através da análise dos gráficos obtidos nas Figuras 11 e 12, visto que os pontos associados às temperaturas reais e preditas estão mais próximos uns dos outros na Figura 12.

Figura 11: Comparação entre a temperatura de saída da água (°C) real e obtida com a predição através da RNA com uma camada oculta de 5 nós internos para dados de teste



Fonte: Elaborado pelo autor.

Figura 12: Comparação entre a temperatura de saída da água (°C) real e obtida com a predição através da RNA com duas camadas ocultas de 5 e 4 nós internos para dados de teste



Fonte: Elaborado pelo autor.

4.2 Resultados das RNA com regularização de dados

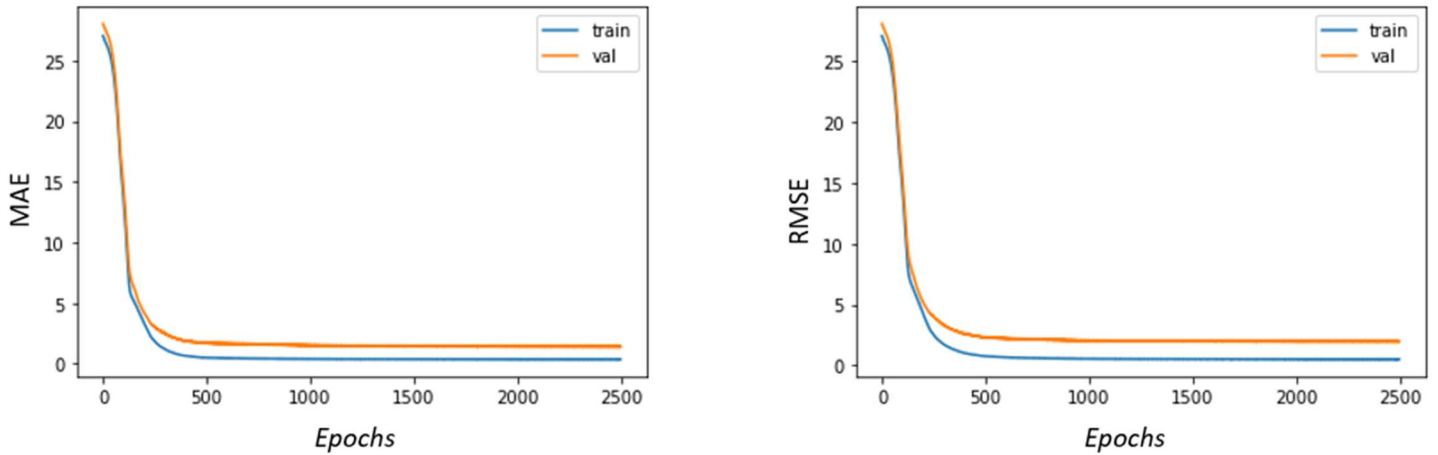
Para cada topologia de rede, com uma e duas camadas ocultas, foram selecionadas as configurações com os melhores desempenhos de predição em termos das métricas estatísticas MAE e RMSE. Para estas arquiteturas, foi testado o efeito das técnicas de regularização de dados sobre o desempenho das redes e sobre um possível *overfitting*. Com isto, será possível alcançar uma melhor capacidade de generalização da rede, resultando em predições mais próximas da realidade. Analisando-se os resultados apresentados na Tabela 3, é possível perceber que, no caso da RNA com 5 e 4 nós internos, os valores de MAE e RMSE obtidos são superiores aos valores da rede quando não há a utilização de técnicas de regularização de dados (MAE = 1,49 e RMSE = 1,65).

Tabela 3: Resultados obtidos em cada RNA com a aplicação das técnicas de regularização de dados L1, L2 e Dropout.

Nº HL	Nº de Nós	Técnica	Hiperparâmetro (λ ou p)	MAE original	MAE	Δ MAE	RMSE original	RMSE	Δ RMSE
1	5	Dropout	0,1	1,83	3,98	117%	1,98	4,06	105%
1	5	Dropout	0,2	1,83	4,52	147%	1,98	4,58	132%
1	5	Dropout	0,3	1,83	3,30	80%	1,98	3,38	71%
1	5	Dropout	0,6	1,83	4,63	153%	1,98	4,69	137%
1	5	Dropout	0,9	1,83	8,82	381%	1,98	8,86	348%
1	5	L1	0,01	1,83	1,97	7%	1,98	2,11	7%
1	5	L1	0,05	1,83	2,13	16%	1,98	2,26	14%
1	5	L1	0,1	1,83	2,19	19%	1,98	2,32	17%
1	5	L1	0,2	1,83	2,40	31%	1,98	2,52	27%
1	5	L1	0,3	1,83	2,53	38%	1,98	2,65	34%
1	5	L2	0,01	1,83	1,77	-3%	1,98	1,94	-2%
1	5	L2	0,05	1,83	1,97	8%	1,98	2,11	7%
1	5	L2	0,1	1,83	1,95	6%	1,98	2,09	6%
1	5	L2	0,2	1,83	1,92	5%	1,98	2,06	4%
1	5	L2	0,3	1,83	2,07	13%	1,98	2,20	11%
2	5 4	Dropout	0,1	1,49	1,75	17%	1,65	2,01	22%
2	5 4	Dropout	0,2	1,49	5,36	259%	1,65	5,45	231%
2	5 4	Dropout	0,3	1,49	4,01	168%	1,65	4,09	149%
2	5 4	Dropout	0,6	1,49	7,46	399%	1,65	7,51	356%
2	5 4	Dropout	0,9	1,49	10,12	577%	1,65	10,18	518%
2	5 4	L1	0,01	1,49	2,16	44%	1,65	2,30	40%
2	5 4	L1	0,05	1,49	2,07	38%	1,65	2,20	34%
2	5 4	L1	0,1	1,49	2,21	48%	1,65	2,34	42%
2	5 4	L1	0,2	1,49	2,32	55%	1,65	2,44	48%
2	5 4	L1	0,3	1,49	2,41	61%	1,65	2,53	54%
2	5 4	L2	0,01	1,49	2,12	42%	1,65	2,27	38%
2	5 4	L2	0,05	1,49	1,82	22%	1,65	1,96	19%
2	5 4	L2	0,1	1,49	2,09	40%	1,65	2,23	35%
2	5 4	L2	0,2	1,49	2,06	38%	1,65	2,19	33%
2	5 4	L2	0,3	1,49	2,22	48%	1,65	2,34	42%

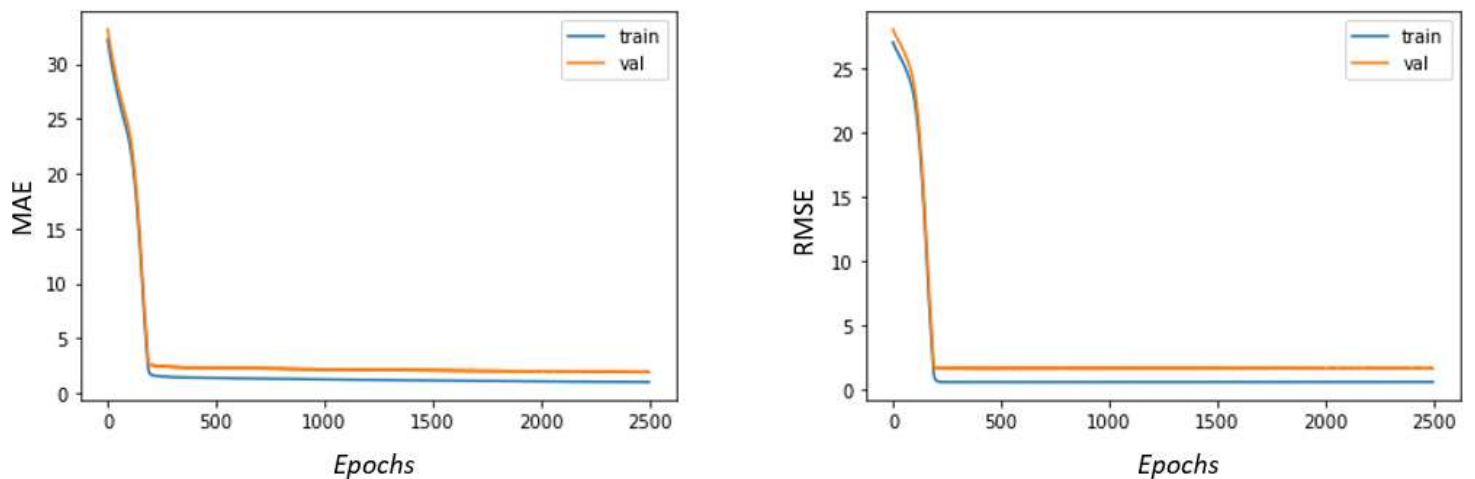
Similarmente, a curva de aprendizado desta arquitetura de rede não apresentou mudanças qualitativas em seu comportamento em termos de sobre-ajuste em nenhum caso testado para nenhuma combinação de técnica de regularização de dados e hiperparâmetro. Este fato pode ser visto na comparação entre a Figura 13, representando as curvas sem a aplicação de técnicas de regularização, com as Figuras 14, 15 e 16.

Figura 13: Curvas de aprendizado obtidas no treinamento, sem regularização, da RNA com duas camadas ocultas com 5 e 4 nós. À esquerda, a curva associada à função perda (MAE) e à direita a curva associada à métrica RMSE.



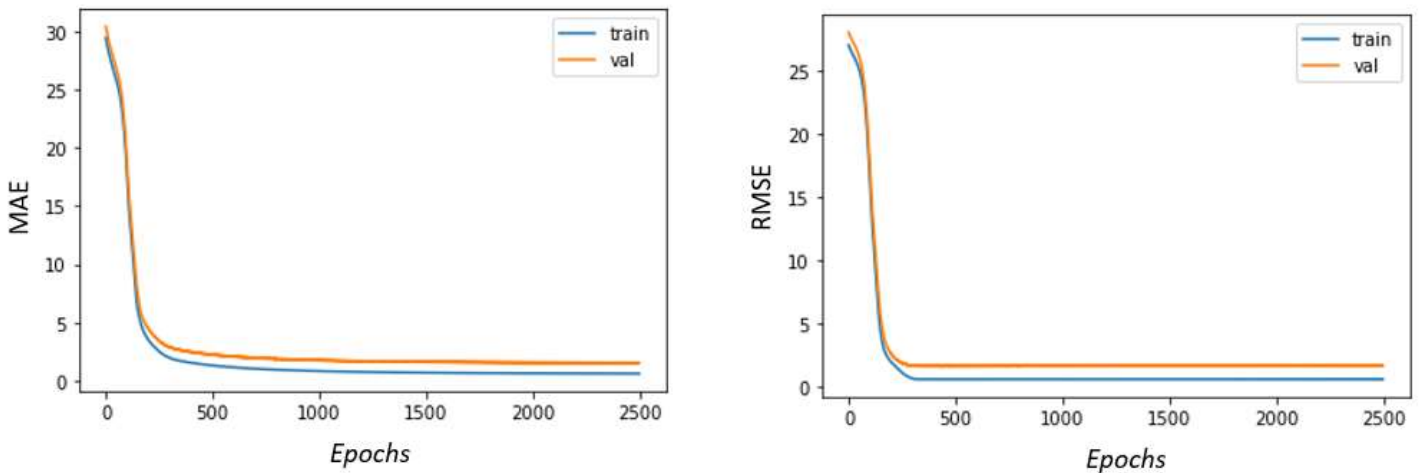
Fonte: Elaborado pelo autor.

Figura 14: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 e 4 nós ocultos utilizando a técnica de regularização de dados L1 com $\lambda = 0,3$.



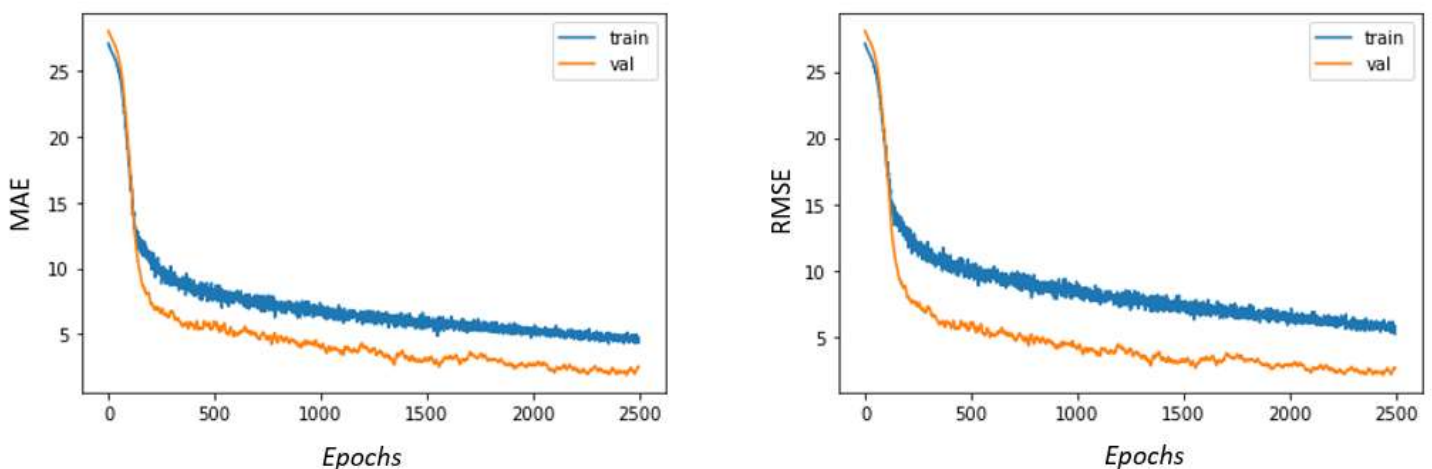
Fonte: Elaborado pelo autor.

Figura 15: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 e 4 nós ocultos utilizando a técnica de regularização de dados L2 com $\lambda = 0,3$.



Fonte: Elaborado pelo autor.

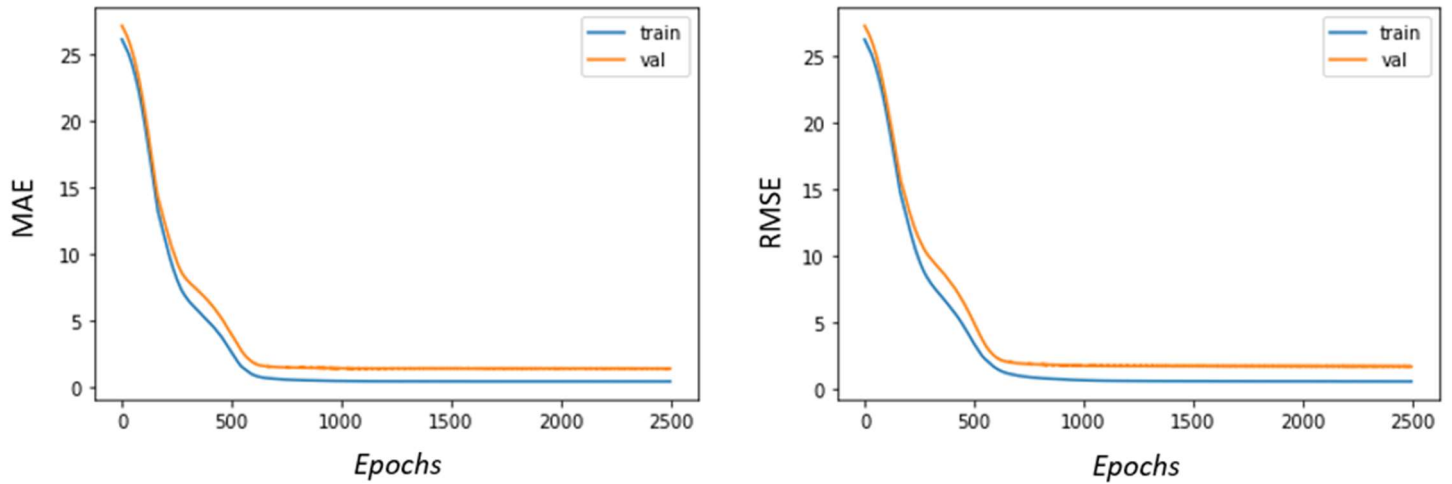
Figura 16: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 e 4 nós ocultos utilizando a técnica de regularização de dados *Dropout* com $p = 0,3$.



Fonte: Elaborado pelo autor.

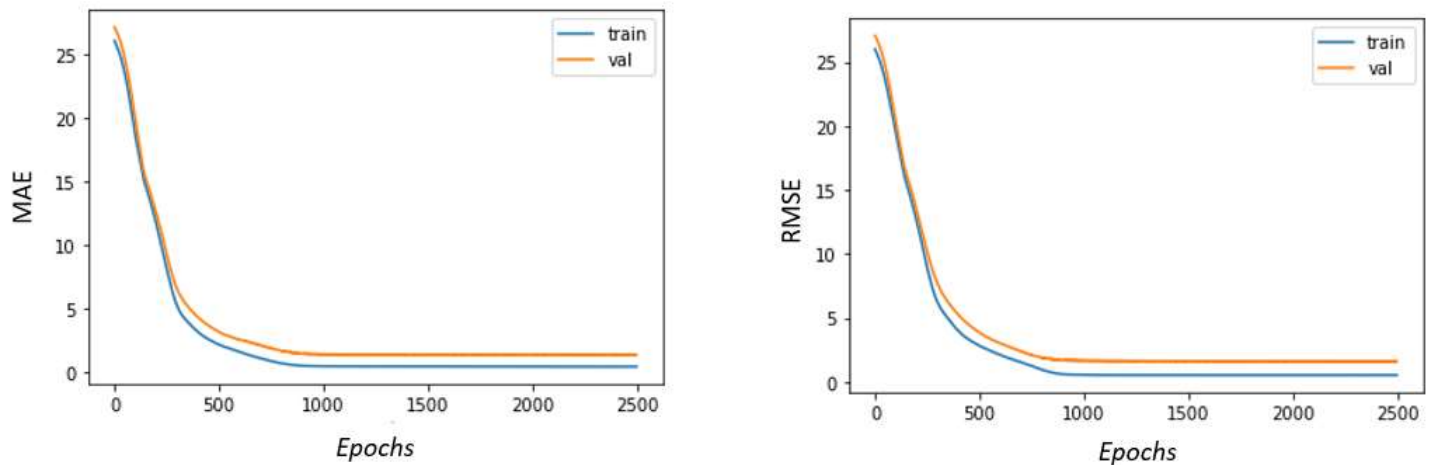
De acordo com a Tabela 3, em relação à RNA com 5 nós na camada oculta, é possível verificar somente um caso no qual ocorreu uma melhoria na capacidade de predição da rede: com a utilização da técnica L2 usando o valor de λ igual a 0,01. O treinamento desta rede resultou em valores menores para as métricas MAE (1,77) e RMSE (1,94) quando comparado à rede original (MAE = 1,83 e RMSE = 1,98). As Figuras 18, 19 e 20 apresentam as curvas de aprendizado resultantes para o treinamento de redes com esta configuração aplicando as técnicas de regularização de dados, enquanto a Figura 17 apresenta as curvas de aprendizado associadas à rede sem a aplicação das técnicas de regularização, para comparação.

Figura 17: Curvas de aprendizado obtidas no treinamento, sem regularização, da RNA com uma camada oculta com 5 nós. À esquerda, a curva associada à função perda (MAE) e à direita a curva associada à métrica RMSE.



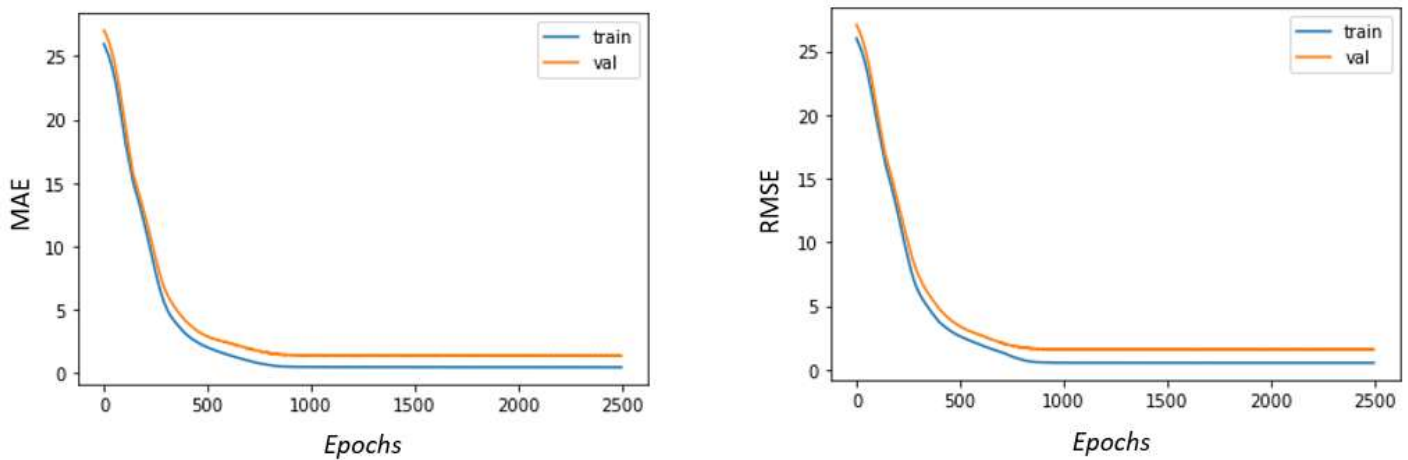
Fonte: Elaborado pelo autor.

Figura 18: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 nós ocultos utilizando a técnica de regularização de dados L1 com $\lambda = 0,01$.



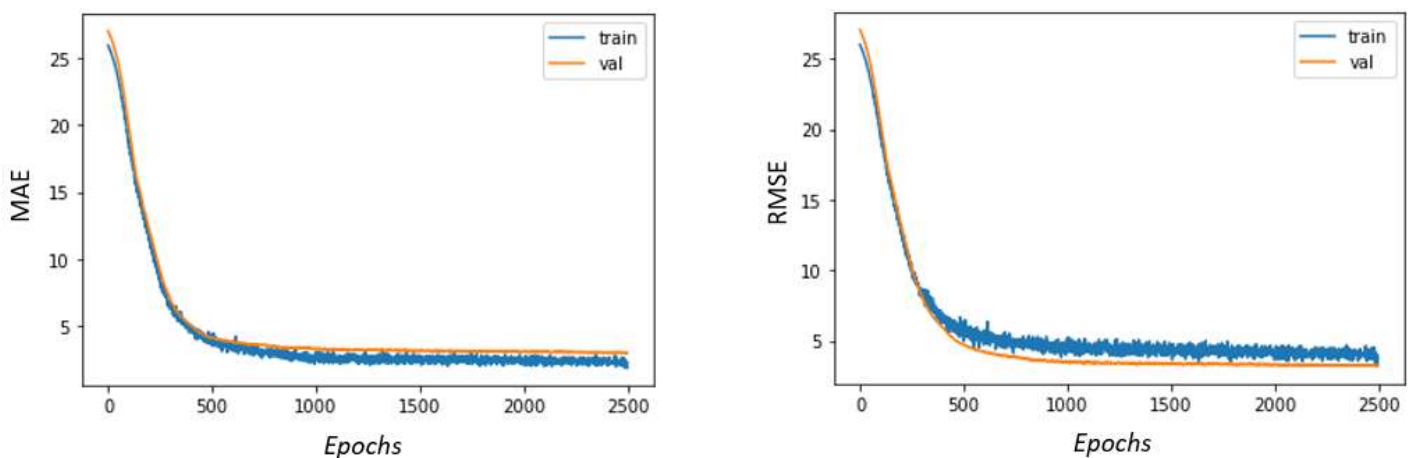
Fonte: Elaborado pelo autor.

Figura 19: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 nós ocultos utilizando a técnica de regularização de dados L2 com $\lambda = 0,01$.



Fonte: Elaborado pelo autor.

Figura 20: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 nós ocultos utilizando a técnica de regularização de dados *Dropout* com $p = 0,1$.



Fonte: Elaborado pelo autor.

Apesar da rede treinada e testada utilizando a técnica de regularização L2 com $\lambda = 0,01$ ter apresentado melhorias nos valores das métricas estatísticas MAE e RMSE, representando uma capacidade de generalização maior, é possível notar, na Figura 19, que a aplicação da regularização não impactou de forma significativa o comportamento da curva de aprendizado. Esta afirmação pode ser justificada em função das características da ocorrência do sobreajuste que, mesmo que em pequena escala, ainda estão presentes, ou seja, ainda é possível perceber, com o progresso do treinamento, valores de MAE e RMSE maiores para o conjunto de dados de validação em relação ao conjunto de dados de treinamento.

Os dados apresentados na Tabela 3 mostram, também, que os resultados obtidos de MAE e RMSE utilizando a técnica de regularização *Dropout* tiveram os piores desempenhos, tanto

para a configuração da rede com uma camada oculta quanto com duas. Ao mesmo tempo, como pode ser visualizado nas Figuras 16 e 20, é possível verificar que o uso da técnica *Dropout* ocasionou ruídos nas curvas de aprendizado. Além disso, as curvas associadas aos dados de treinamento estão acima das curvas associadas aos dados de validação. Também, de forma geral, a técnica de regularização de dados L2 obteve resultados mais satisfatórios para as métricas MAE e RMSE quando comparado com os resultados obtidos utilizando a técnica L1.

4.3 Análise exploratória para o hiperparâmetro λ

Realizou-se uma análise exploratória para o valor do hiperparâmetro λ no caso da RNA de 5 nós na camada interna, caso no qual a técnica L2 obteve um resultado superior ao original, a fim de determinar o valor deste hiperparâmetro que resulta nos menores valores para MAE e RMSE. Tal procedimento foi realizado testando-se diversos valores de λ próximos a 0,01, comparando-se valores de MAE e RMSE obtidos em cada caso. A Tabela 4 apresenta os resultados obtidos com a análise exploratória. Assim, é possível verificar que o valor de λ que obteve o melhor resultado na análise exploratória foi igual a 0,01597, com um valor de MAE = 1,66 e de RMSE = 1,83.

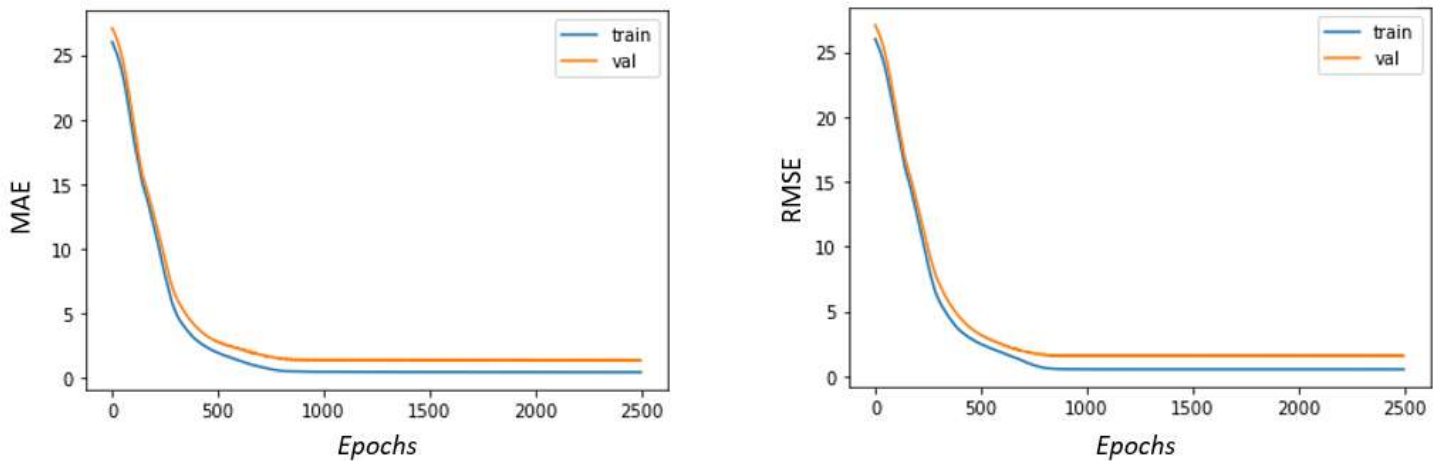
Tabela 4: Resultados da análise exploratória para o valor do hiperparâmetro λ , utilizando a técnica de regularização L2.

λ	MAE	RMSE
0,0095	1,82	1,99
0,01	1,77	1,94
0,015	1,75	1,92
0,0159	1,76	1,93
0,01595	1,72	1,89
0,01596	1,73	1,90
0,01597	1,66	1,83
0,01598	1,70	1,87
0,016	1,68	1,85
0,0161	1,70	1,87
0,0165	1,72	1,89
0,017	1,69	1,86

As Figuras 21 e 22 apresentam, respectivamente, as curvas de aprendizado resultantes do treinamento e teste da rede utilizando a técnica L2 com $\lambda = 0,01597$ e o gráfico que compara as temperaturas de saída da água preditas com as reais. Com elas, pode-se constatar que a diferença entre as curvas referentes aos dados de treinamento e de validação continua semelhante à situação anterior. Além disso, a comparação das temperaturas de saída da água preditas e reais, apesar de ser semelhante à da rede original sem regularização de dados, demonstra menores diferenças entre as temperaturas preditas e reais. Com esta estrutura de rede final obtida, é interessante comparar os resultados de MAE e RMSE obtidos com os valores associados à rede treinada sem a separação dos dados em conjuntos diferentes

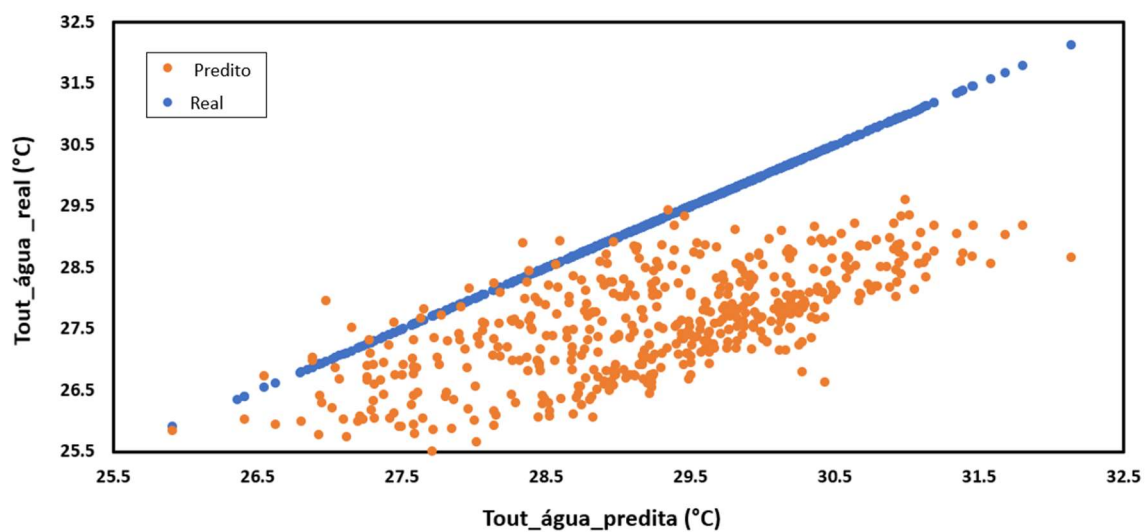
através do algoritmo *K-Means*, ou seja, destinando aleatoriamente os dados nos respectivos conjuntos de treinamento, teste e validação. Assim, realizando a separação dos dados aleatoriamente, foi possível obter, para a rede com 1 camada oculta de 5 nós internos, valores de MAE = 0,85 e RMSE = 1,09.

Figura 21: Curvas de aprendizado resultantes do treinamento e teste da RNA com 5 nós ocultos utilizando a técnica de regularização de dados L2 com $\lambda = 0,01597$



Fonte: Elaborado pelo autor.

Figura 22: Comparação entre a temperatura de saída da água (°C) real e obtida com a predição através da RNA (5 nós ocultos, regularização L2 com $\lambda = 0,01597$) para dados de teste.



Fonte: Elaborado pelo autor.

5 Conclusões e Trabalhos Futuros

O estudo realizado teve como principal objetivo o desenvolvimento de Redes Neurais Artificiais e a comparação de técnicas de Regularização de Dados aplicados à predição da temperatura de saída da água em uma torre de resfriamento. Inicialmente, foi possível concluir que, sem a aplicação de regularização de dados, as redes com duas camadas ocultas tiveram, de forma geral, melhores resultados que as redes com uma camada oculta quando comparadas em relação aos valores de MAE e RMSE. Ainda, entre as redes inicialmente testadas sem a regularização, a que obteve os melhores resultados de predição foi a rede com duas camadas ocultas com 5 e 4 nós ocultos (MAE = 1,49 e RMSE = 1,65).

As técnicas de regularização de dados utilizadas (L1, L2 e *Dropout* com seus respectivos hiperparâmetros) não apresentaram impactos significativos na diminuição do efeito de *overfitting*, mesmo que presente em pequena escala. Tal afirmação se baseia no comportamento das curvas de aprendizado resultantes, que não tiveram mudanças perceptíveis, e dos valores de MAE e RMSE resultantes da aplicação das redes treinadas com regularização de dados aos dados de teste, os quais foram, em sua grande maioria, maiores que os originais, principalmente os valores para a técnica *Dropout*. Contudo, a técnica L2 com $\lambda = 0,01$ no caso da rede com uma camada oculta e 5 nós ocultos demonstrou resultados melhores (MAE = 1,77 e RMSE = 1,94) que os da rede sem regularização (MAE = 1,83 e RMSE = 1,98). Tal resultado pode ser refinado através de uma análise exploratória que proporcionou resultados ainda melhores (MAE = 1,66 e RMSE = 1,83).

Entretanto, a análise da curva de aprendizado resultante para a rede com 5 nós ocultos utilizando a regularização L2 com $\lambda = 0,01$ também não demonstrou impactos na diminuição do efeito de *overfitting* existente, tendo em vista a manutenção da distância entre as curvas associadas aos dados de treino e de validação ao longo das épocas.

A observação de que o impacto das técnicas de regularização não pôde ser notado significativamente na diminuição do *overfitting* pode estar associado ao fato de a base de dados históricos da torre de resfriamento ser relativamente pequena, o que diminui as chances de ocorrência do sobre-ajuste, juntamente com a sua intensidade. Uma base de dados maior aumenta a variedade de dados, de forma que, dependendo da forma que a etapa de treinamento é feita, pode aumentar a dificuldade de generalização para dados distintos. Nesse sentido, é interessante que, para estudos futuros sejam usadas bases de dados maiores, principalmente quando se deseja estudar a aplicabilidade de técnicas de regularização de dados, como L1, L2 e *Dropout*, em redes que sofrem com o efeito do *overfitting*.

REFERÊNCIAS

ALVES, Rita Maria de Brito; NASCIMENTO, Cláudio Augusto Oller do. Otimização de um processo industrial de produção de isopreno via redes neurais, p. 8–30, 2003. Tese de Doutorado. Universidade de São Paulo (USP). Escola Politécnica (EP/BC).

BADOLA, Akshay; NAIR, Vineet Padmanabhan; LAL, Rajendra Prasad. An analysis of regularization methods in deep neural networks. In: **2020 IEEE 17th India Council International Conference (INDICON)**. IEEE, 2020. p. 1-6.

BOYKO, Nataliya; MOROZ, Olena. Comparative Analysis of Regression Regularization Methods for Life Expectancy Prediction. In: **MoMLet+ DS**. 2021. p. 310-326.

DEMIR-KAVUK, Ozgur et al. Prediction using step-wise L1, L2 regularization and feature selection for small data sets with large number of features. **BMC bioinformatics**, v. 12, n. 1, p. 1-10, 2011.

DOBBELAERE, Maarten R. et al. Machine learning in chemical engineering: strengths, weaknesses, opportunities, and threats. **Engineering**, v. 7, n. 9, p. 1201-1211, 2021.

DOMHAN, Tobias; SPRINGENBERG, Jost Tobias; HUTTER, Frank. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In: **Twenty-fourth international joint conference on artificial intelligence**. 2015.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. MIT press, p.98-155, 2016.

GOVENDER, Paulene; SIVAKUMAR, Venkataraman. Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019). **Atmospheric pollution research**, v. 11, n. 1, p. 40-56, 2020.

HAYKIN, Simon. **Redes neurais: princípios e prática**. Bookman Editora, p. 27-69, 2001.

HIMMELBLAU, David M. Applications of artificial neural networks in chemical engineering. **Korean journal of chemical engineering**, v. 17, n. 4, p. 373-392, 2000.

HWANG, Jun Kwon et al. A novel hybrid deep neural network model to predict the refrigerant charge amount of heat pumps. **Sustainability**, v. 12, n. 7, p. 2914, 2020.

Investigating Underfitting and Overfitting – Medium. Disponível em: <<https://medium.com/geekculture/investigating-underfitting-and-overfitting-70382835e45c>>. Acesso em: 07 jun. 2022.

LEE, Hagyeong; SONG, Jongwoo. Introduction to convolutional neural network using Keras; an understanding from a statistician. **Communications for Statistical Applications and Methods**, v. 26, n. 6, p. 591-610, 2019.

MILLÉO, Carla. Emprego de redes neurais artificiais na previsão climática de temperatura e precipitação no Estado do Paraná, p. 24-45, 2020. Dissertação de Mestrado. Universidade Federal do Paraná. Setor de Tecnologia. Programa de Pós-Graduação em Engenharia Ambiental.

Neural Networks Models in R – Datacamp. Disponível em:

<<https://www.datacamp.com/tutorial/neural-network-models-r>>. Acesso em: 23 mai. 2022.

NUSRAT, Ismoilov; JANG, Sung-Bong. A comparison of regularization techniques in deep neural networks. **Symmetry**, v. 10, n. 11, p. 648, 2018.

Overfitting and Underfitting – Kaggle. Disponível em:

<<https://www.kaggle.com/code/ryanholbrook/overfitting-and-underfitting/tutorial>>. Acesso em: 11 jun. 2022.

PERLICH, Claudia. Learning Curves in Machine Learning, p. 1-7, 2010.

PHAISANGITTISAGUL, Ekachai. An analysis of the regularization between L2 and dropout in single hidden layer neural network. In: **2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)**. IEEE, 2016. p. 174-179.

Python Numerical Methods – Berkeley. Disponível em:

<<https://pythonnumericalmethods.berkeley.edu/notebooks/chapter25.01-Concept-of-Machine-Learning.html>>. Acesso em: 15 mai. 2022.

SINGH, Amit Kumar; SINGH, H. P.; MISHRA, Sachin. Validation of ann-based model for binary distillation column. In: **Proceeding of International Conference on Intelligent Communication, Control and Devices**. Springer, Singapore, 2017. p. 235-242.

SIQUEIRA, Felipe Macedo Freitas. Estratégia de otimização de processos com o uso de redes neurais artificiais, p. 22-41, 2021. Dissertação de Mestrado. Universidade Federal Fluminense, Niterói.

SRIVASTAVA, Nitish et al. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, v. 15, n. 1, p. 1929-1958, 2014.

TARCA, Laurentiu A.; GRANDJEAN, Bernard PA; LARACHI, Faical. Reinforcing the phenomenological consistency in artificial neural network modeling of multiphase reactors. **Chemical Engineering and Processing: Process Intensification**, v. 42, n. 8-9, p. 653-662, 2003.

VENKATASUBRAMANIAN, Venkat. The promise of artificial intelligence in chemical engineering: Is it here, finally?. **AIChE Journal**, v. 65, n. 2, p. 466-478, 2019.

WAXENEGGER-WILFING, Günther et al. Heat transfer prediction for methane in regenerative cooling channels with neural networks. **Journal of Thermophysics and Heat Transfer**, v. 34, n. 2, p. 347-357, 2020.

ZHANG, J. et al. Prediction of polymer quality in batch polymerisation reactors using robust neural networks. **Chemical Engineering Journal**, v. 69, n. 2, p. 135-143, 1998.

APÊNDICE A

```
#__Clusterizacao através de K-Means__
import pandas as pd
import tensorflow as tf
from tensorflow import keras
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from matplotlib import pyplot as plt
from keras.models import Sequential
from keras.layers import Dense
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.cluster import KMeans
%matplotlib inline

df = pd.read_csv('Dados_Python_V3.csv')

base_teste = pd.DataFrame(df, columns=['T_bulbo_seco_ar',
'T_agua_fria'])

kmeans = KMeans(n_clusters=3, init='k-means++',
random_state=0).fit(base_teste)

import seaborn as sns
import matplotlib.pyplot as plt
sns.scatterplot(data=base_teste, x="T_bulbo_seco_ar",
y="T_agua_fria", hue=kmeans.labels_)

plt.show()
kmeans.labels_.shape

pd.DataFrame(kmeans.labels_).to_csv('KMeans_Labels.csv')

#__Construcao das RNA sem regularização de dados__

import pandas as pd
import tensorflow as tf
from tensorflow import keras
import numpy as np
from sklearn.model_selection import train_test_split
```



```
from sklearn.preprocessing import StandardScaler
from matplotlib import pyplot as plt
from keras.models import Sequential
from keras.layers import Dense
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
np.random.seed(42)
tf.random.set_seed(42)
%matplotlib inline

#Usando pandas, pego os dados que estão no formato .csv
df_validacao =
pd.read_csv('Dados_Python_5_K_Means_0_Testes.csv')
df_treino = pd.read_csv('Dados_Python_5_K_Means_2_Treino.csv')
df_teste =
pd.read_csv('Dados_Python_5_K_Means_1_Validacao.csv')

x_input_teste = df_teste.drop(['T_agua_fria'], axis=1)
y_output_teste = df_teste[['T_agua_fria']]

x_input_treino = df_treino.drop(['T_agua_fria'], axis=1)
y_output_treino = df_treino[['T_agua_fria']]

x_input_validacao = df_validacao.drop(['T_agua_fria'], axis=1)
y_output_validacao = df_validacao[['T_agua_fria']]

#Nos dados separados acima, realizo a normalização dos X_train
e X_test - utilização da z-score normalization
def scale_datasets(x_input_treino, x_input_teste):
    standard_scaler = StandardScaler()
    X_train_scaled = pd.DataFrame(
        standard_scaler.fit_transform(x_input_treino),
        columns= x_input_treino.columns
    )
    X_test_scaled = pd.DataFrame(
        standard_scaler.transform(x_input_teste),
        columns = x_input_teste.columns
    )
    return X_train_scaled, X_test_scaled

X_train_scaled, X_test_scaled = scale_datasets(x_input_treino,
x_input_teste)

def scale_datasets_v(x_input_validacao):
    standard_scaler = StandardScaler()
    X_validation_scaled = pd.DataFrame(
        standard_scaler.fit_transform(x_input_validacao),
        columns= x_input_validacao.columns
    )
    return X_validation_scaled
X_validation_scaled = scale_datasets_v(x_input_validacao)
```

```
#Convertendo os intervalos de dados tratados acima em np.array
para utilização do Keras
X_train_scaled = np.array(X_train_scaled)
X_test_scaled = np.array(X_test_scaled)
X_validation_scaled = np.array(X_validation_scaled)
y_train = np.array(y_output_treino)
y_test = np.array(y_output_teste)
y_validation = np.array(y_output_validacao)

#Início construção da Rede Neural Artificial*semregularização*
model = Sequential()
model.add(Dense(5, input_dim=6, activation="relu"))
model.add(Dense(4, activation="relu"))
model.add(Dense(1))

#Compila o modelo definido anteriormente
model.compile(loss='mean_absolute_error', optimizer='adam',
metrics=[tf.keras.metrics.RootMeanSquaredError()])
#model.summary()
#Aplica o modelo aos dados de treino para setar os pesos
history = model.fit(X_train_scaled, y_train, epochs=2500,
batch_size=64, verbose=1,
validation_data=(X_validation_scaled, y_validation), shuffle =
False)

#Aplica a rede obtida aos dados de teste
y_predict = model.predict(X_test_scaled)
print("T_agua_fria MAE:%.4f" %
mean_absolute_error(y_test[:,0], y_predict[:,0]))
print("T_agua_fria RMSE:%.4f" %
mean_squared_error(y_test[:,0], y_predict[:,0],
squared=False))

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper right')
plt.show()

plt.plot(history.history['root_mean_squared_error'])
plt.plot(history.history['val_root_mean_squared_error'])
plt.title('model root_mean_squared_error')
plt.ylabel('root_mean_squared_error')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper right')
plt.show()
```

```
pd.DataFrame(y_predict).to_csv('sample_prediction.csv')
pd.DataFrame(y_test).to_csv('sample_test.csv')

#___Construcao das RNA com regularização de dados___

. . .

#Início da construção da Rede Neural Artificial
model = Sequential()
model.add(Dense(5, input_dim=6, activation="relu",
kernel_regularizer=keras.regularizers.l2(0.01597)))
#model.add(Dropout(0.9))
#model.add(Dense(4, activation="relu"))#,
kernel_regularizer=keras.regularizers.l2(0.3))
#model.add(Dropout(0.9))
model.add(Dense(1))

#Compila o modelo definido anteriormente
model.compile(loss='mean_absolute_error', optimizer='adam',
metrics=[tf.keras.metrics.RootMeanSquaredError()])

#Aplica o modelo aos dados de treino para setar os pesos
history = model.fit(X_train_scaled, y_train, epochs=2500,
batch_size=64, verbose=1,
validation_data=(X_validation_scaled, y_validation), shuffle =
False)

#Aplica a rede obtida aos dados de teste
y_predict = model.predict(X_test_scaled)
print("T_agua_fria MAE:%.4f" %
mean_absolute_error(y_test[:,0], y_predict[:,0]))
print("T_agua_fria RMSE:%.4f" %
mean_squared_error(y_test[:,0], y_predict[:,0],
squared=False))

. . .
```