



APLICAÇÃO DE REDES NEURAIS PARA INFERÊNCIA DE TEMPERATURAS HISTÓRICAS NA USINA NUCLEAR ANGRA 1

Fernando Henrique Pereira Cardozo

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Nuclear, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Nuclear.

Orientador: Roberto Schirru

Rio de Janeiro
Março de 2022

APLICAÇÃO DE REDES NEURAIS PARA INFERÊNCIA DE
TEMPERATURAS HISTÓRICAS NA USINA NUCLEAR ANGRA 1

Fernando Henrique Pereira Cardozo

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO
ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE
ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU
DE MESTRE EM CIÊNCIAS EM ENGENHARIA NUCLEAR.

Orientador: Roberto Schirru

Aprovada por: Prof. Roberto Schirru

Prof. Victor Henrique Cabral Pinheiro

Prof. Paulo Victor Rodrigues de Carvalho

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2022

Cardozo, Fernando Henrique Pereira

Aplicação de redes neurais para inferência de temperaturas históricas na usina nuclear Angra 1/Fernando Henrique Pereira Cardozo. – Rio de Janeiro: UFRJ/COPPE, 2022.

XIV, 76 p.: il.; 29,7cm.

Orientador: Roberto Schirru

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Nuclear, 2022.

Referências Bibliográficas: p. 73 – 76.

1. Energia Nuclear.
 2. Inteligência Artificial.
 3. Redes Neurais Artificiais.
- I. Schirru, Roberto.
II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Nuclear. III. Título.

*Dedico este trabalho à minha
família que esteve comigo em
todas as etapas de meus estudos.*

Agradecimentos

Aos meus pais, Henrique e Cristina e aos meus irmãos, Augusto e Marina, por todo o apoio e suporte durante toda a jornada.

À todos os meus amigos, por estarem sempre comigo em todas as etapas e pela paciência nos momentos em que não pude dar a devida atenção devido aos estudos.

Ao meu orientador Prof. Roberto Schirru, por toda ajuda, debates e ensinamentos ao longo de todo o curso de mestrado.

À Equipe do Laboratório de Monitoração de Processos, com quem estive trabalhando durante o desenvolvimento deste trabalho.

Aos membros da banca por aceitarem participar da banca, lendo o trabalho e dando sugestões de melhorias.

*"Ninguém é tão ignorante
que não tenha algo a ensi-
nar. Ninguém é tão sábio
que não tenha algo a apren-
der."– Blaise Pascal*

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

APLICAÇÃO DE REDES NEURAIIS PARA INFERÊNCIA DE TEMPERATURAS HISTÓRICAS NA USINA NUCLEAR ANGRA 1

Fernando Henrique Pereira Cardozo

Março/2022

Orientador: Roberto Schirru

Programa: Engenharia Nuclear

Apresenta-se, neste trabalho, uma proposta de solução para a ausência do histórico de temperaturas na usina nuclear Angra 1, que deseja solicitar a extensão de sua vida útil.

Equipamentos no interior do prédio de contenção do reator estão suscetíveis a elevadas temperaturas, fazendo com que sua vida útil qualificada possa ser diferente daquela de base de projeto.

Estudos do envelhecimento dos equipamentos são necessários para a elaboração do programa de qualificação ambiental, documento este necessário à solicitação de extensão de vida útil de uma usina nuclear. Para elaboração deste estudo é necessário conhecimento de um histórico de temperaturas à qual os equipamentos foram submetidos ao longo dos anos de operação.

Um problema comum que é encontrado no momento da realização do estudo de envelhecimento é a ausência desse histórico de temperaturas em posições estratégicas ao estudo de envelhecimento.

Este trabalho propõe uma metodologia de inferência de temperaturas, simulando detectores históricos através do uso de redes neurais artificiais. A metodologia utiliza medições dos detectores que foram instalados em posições estratégicas na usina a partir do ano de 2015 em conjunto com medições de temperatura do Sistema Integrado de Computadores de Angra.

São treinadas redes neurais DNN e LSTM com o objetivo tanto de fazer a inferência das temperaturas quanto com o objetivo de comparar a performance de cada arquitetura no conjunto de dados.

Ambas arquiteturas apresentam resultados satisfatórios ao final do trabalho, com a LSTM apresentando um resultado 25% melhor do que a DNN.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

APPLICATION OF NEURAL NETWORK TO HISTORY TEMPERATURE
INFERENCE IN ANGRA 1 NUCLEAR POWER PLANT

Fernando Henrique Pereira Cardozo

March/2022

Advisor: Roberto Schirru

Department: Nuclear Engineering

In this work, a solution proposal is presented for the absence of temperature history in Angra 1 Nuclear Power Plant, which wish to request its long-term operation.

Equipment in the reactor containment building are susceptible to high temperatures, making its lifetime qualified to be different from that of project base.

Ageing studies of this equipment are necessary to elaborate the environmental qualification program, document needed to request the long-term operation license. To elaborate this studies it is necessary a history of temperature which this equipment has been submitted over the years of operation.

A usual problem found at the moment of ageing study is the absence of temperature history over the years in strategical positions to the ageing study.

This work proposes a temperature inference methodology, simulating historical detectors through artificial neural networks. The methodology uses measurements of detectors installed since year 2015 together with measurements of the *Sistema Integrado de Computadores de Angra*.

Two architectures of neural networks are trained, DNN and LSTM, with two goals, making the inference of temperature and to compare which architecture performs better in the database.

Both architectures present satisfactory results at the end of the work, with LSTM presenting a result 25% better than DNN.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Abreviaturas	xiv
1 Introdução	1
2 Fundamentos Teóricos	7
2.1 Segurança Nuclear	7
2.2 Extensão de Vida Útil	10
2.2.1 Programa de Qualificação de Equipamentos	12
2.3 Modelo de Envelhecimento de Arrhenius	15
2.4 Fundamentação Estatística	16
2.4.1 Distribuição Gaussiana	17
2.4.2 Medidas de Correlação	20
2.5 Redes Neurais Artificiais	21
2.5.1 História das Redes Neurais Artificiais	21
2.5.2 O Neurônio Biológico	23
2.5.3 O Neurônio Matemático	24
2.5.4 O Perceptron	25
2.5.5 Redes Neurais Profundas	27
3 O Problema e a Metodologia Proposta	42
3.1 Seleção de Variáveis	43
3.2 Pré-Processamento dos dados	44
3.3 Modelagem da Rede Neural	46
3.3.1 Modelagem da DNN	47
3.4 Modelagem Pós-Análise de Correlações	56
3.4.1 Pré-Processamento	56

4	Resultados	59
4.1	Treinamento DNN	59
4.1.1	Resultados DNN	59
4.2	Treinamento LSTM	61
4.2.1	Resultados LSTM	63
5	Conclusão	70
	Referências Bibliográficas	73

Lista de Figuras

2.1	Representação do Neurônio Biológico (DSA, 2018).	23
2.2	Representação do Neurônio Matemático DSA, 2018).	24
2.3	Multiplicação Matricial que ocorre entre as entradas X e os pesos W DSA, 2018).	25
2.4	Representação do Perceptron.	26
2.5	Representação de uma Rede Neural Simples, à esquerda, e de uma rede neural profunda, à direita (DSA, 2018).	28
2.6	Representação da Descida do Gradient para uma variável.	29
2.7	Rede Neural Profunda Simplificada.	33
2.8	Derivada da Função Sigmoide.	35
2.9	Célula LSTM (OLAH, 2015).	37
2.10	Portão de Esquecimento, Célula LSTM (OLAH, 2015).	38
2.11	Portão de Entrada, Célula LSTM (OLAH, 2015).	39
2.12	Portão de Saída, Célula LSTM (OLAH, 2015).	40
3.1	Sensor SICA-38. À esquerda, gráfico antes do tratamento de <i>outlier</i> , à direita, após o tratamento.	45
3.2	Sensor SICA-109. À esquerda, gráfico antes do tratamento de <i>outlier</i> , à direita, após o tratamento.	46
3.3	Representação das entradas e saídas da rede neural na metodologia proposta.	46
3.4	Previsão no ciclo 2017-2018 para o Sensor-38.	51
3.5	Previsão no ciclo 2017-2018 para o Sensor-66.	51
3.6	Previsão no ciclo 2017-2018 para o Sensor-20.	52
3.7	Previsão no ciclo 2017-2018 para o Sensor-33.	52
3.8	Comparação entre a distribuição de temperatura do Sensor-16 com o SICA-67.	54
3.9	Comparação entre a distribuição de temperatura do Sensor-76 com o sensor SICA-123.	54
3.10	Comparação entre a distribuição de temperatura do Sensor-20 com o sensor SICA-24.	55

3.11	Para 2018-2019. À Esquerda: distribuição de Frequência de Temperaturas do Delta entre o Sensor-16 e o SICA-67. À Direita: distribuição do Delta no tempo.	57
3.12	Para 2017-2018. À Esquerda: distribuição de Frequência de Temperaturas do Delta entre o Sensor-16 e o SICA-67. À Direita: distribuição do Delta no tempo.	57
4.1	Comparação entre o valor real para o Sensor-16 e suas previsões pela LSTM e pela DNN.	65
4.2	Comparação entre o valor real para o Sensor-30 e suas previsões pela LSTM e pela DNN.	66
4.3	Comparação entre o valor real para o Sensor-63 e suas previsões pela LSTM e pela DNN.	66

Lista de Tabelas

3.1	Parâmetros Utilizados no melhor modelo de DNN.	48
3.2	Erros do Modelo DNN.	48
3.3	Estatísticas do Modelo DNN.	49
4.1	Parâmetros Utilizados no melhor modelo de DNN, após análise das correlações.	60
4.2	Erros do Modelo DNN após análise de correlação.	60
4.3	Estatísticas do Modelo DNN após análise de correlação.	61
4.4	Séries de valores de entrada no treinamento da LSTM.	62
4.5	Parâmetros Utilizados no melhor modelo de LSTM, após análise das correlações.	63
4.6	Erros do Modelo LSTM após análise de correlação.	64
4.7	Estatísticas do Modelo LSTM após análise de correlação.	64
4.8	Erro quadrático médio na inferência para o ciclo de operação 2017-2018.	67
4.9	Estatísticas das previsões.	67

Lista de Abreviaturas

CFR	<i>Code of Federal Regulation</i> , p. 11
CNEN	Comissão Nacional de Energia Nuclear, p. 11
DNN	<i>Deep Neural Network</i> , p. 28
EUA	Estados Unidos da América, p. 2
IA	Inteligência Artificial, p. 22
IEEE	<i>Institute of Electrical and Electronic Engineer's</i> , p. 22
LSTM	<i>Long-Short Term Memory</i> , p. 36
MLP	<i>MultiLayers Perceptrons</i> , p. 27
MSE	<i>Mean Squared Error</i> , p. 28
MW	<i>Megawatt</i> , p. 2
NRC	<i>Nuclear Regulatory Comission</i> , p. 10
PQAEE	Programa de Qualificação Ambiental de Equipamentos Eléctricos, p. 4
RPS	Revisão Periódica de Segurança, p. 8
SICA	Sistema Integrado de Computadores de Angra, p. 3

Capítulo 1

Introdução

A demanda energética em todo o mundo acompanha o crescimento da globalização, tendo isto em vista é cada vez mais importante falar e entender sobre geração de energia e alternativas diferentes às tradicionais usinas termoeletricas.

As usinas termonucleares tem um princípio de geração de energia semelhante ao das usinas termoeletricas, onde turbinas no circuito secundário recebem vapor d'água, produzido no circuito primário, e alimentam geradores elétricos, onde a energia mecânica do vapor é transformada em energia elétrica através das bobinas dos geradores.

A grande diferença entre os dois formatos de usina é que enquanto nas termoeletricas esse vapor no circuito primário é gerado através da queima de combustíveis fósseis, que possuem alta emissão de gases de efeito estufa para a atmosfera, nas nucleares o vapor d'água é proveniente das reações de fissão nuclear do elemento combustível, sendo o elemento combustível mais comum nas usinas ao redor do mundo o Urânio-238.

As usinas nucleares são referência em cultura de segurança. Suas construções, operações e descomissionamentos devem atender rígidos critérios de segurança para que sejam garantidas a operação segura, a prevenção de acidentes e a mitigação desses acidentes em caso de ocorrência, dessa forma é garantida a segurança dos operadores e de toda a população.

As usinas nucleares são projetadas para uma vida útil de 40 anos, dentro dos quais é garantida a confiabilidade de todos os componentes e sistemas principais, de segurança e auxiliares. Como em toda indústria os equipamentos sofrem envelhecimento com seu uso e diante disso são necessários estudos para entender e prever esse envelhecimento com o propósito de que sejam feitas manutenções

preventivas e trocas de equipamentos e sistemas de modo que a usina permaneça operando sem falhas.

Conforme uma usina se aproxima do fim de sua vida útil a preocupação passa a ser seu descomissionamento, porém uma alternativa que se provou viável na grande maioria das usinas ao redor do mundo foi a extensão de vida útil, onde são feitos estudos de viabilidade, que levam em conta envelhecimento dos equipamentos e sistemas, custos de manutenção e geração de receita futura. Através desses estudos é possível saber se ela pode, caso todos os requisitos de segurança estejam dentro das especificações, e deve, caso haja potencial de geração de receita financeira para a operadora e energia elétrica para a rede nacional, continuar operando.

No ano de 2014, 74 das 100 usinas nucleares nos EUA tinham recebido licenciamento para operação de longo prazo, ou seja, além dos 40 anos de projeto, enquanto outras 7 usinas estudavam a possibilidade de solicitar ao órgão regulador a extensão. Na Europa, à exceção da Rússia, haviam 151 usinas em operação e dentre elas 66 já estavam operando há mais de 30 anos.

A usina Angra 1, por ter seu prazo de operação até o ano de 2024 começou estudos para a extensão da vida útil no ano de 2015, porém desde 2009 quando foi feita a troca de um dos geradores de vapor, são feitos investimentos com esse objetivo. Cerca de R\$ 1 bilhão estão sendo investidos no projeto de extensão da vida útil da usina, para que ela opere por mais 20 anos, além do seu tempo de projeto. Esse investimento significa um retorno em geração de energia de 640 MW de energia elétrica por mais 20 anos para a rede nacional, mostrando que a extensão de vida útil de uma usina nuclear é uma forma barata de manter a geração de energia (DE SOUZA,2019).

Muita coisa na parte de regulamentação na área nuclear mudou nos últimos 35 anos. Quando Angra 1 entrou em operação, em 1985, não era uma exigência a elaboração de um programa de qualificação ambiental, sendo assim, essa usina não possui tal plano. Conforme o fim da vida útil da usina se aproxima, no ano de 2024 e uma operação de longo prazo se mostra interessante, torna-se necessária a elaboração de um programa de qualificação ambiental, como requisito indispensável ao estudo de viabilidade da extensão de sua vida útil.

O programa de qualificação ambiental visa garantir que as estruturas, componentes e sistemas da usina tenham condições de operar em condições normais, minimizando a possibilidade de acidentes, mitigando as consequências de possíveis

acidentes e impedindo a liberação de material radioativo para o meio ambiente. Essas funções são cumpridas, garantindo que os sistemas da usina sejam capazes de garantir a integridade das barreiras de segurança, barreiras essas que impedem a liberação de material radioativo para fora do prédio de contenção do reator.

Dentro da necessidade da elaboração de um programa de qualificação ambiental, é necessário garantir a integridade dos equipamentos, sistemas e estruturas da usina e para isso é necessário um conhecimento do histórico de temperaturas à qual esses componentes estiveram expostos ao longo dos anos de operação.

O modelo de envelhecimento de Arrhenius (DAVID,1992) correlaciona o tempo útil de um equipamento com a temperatura à qual ele esteve exposto ao longo dos seus anos de operação, fazendo um cálculo de seu envelhecimento e do tempo de vida útil ainda disponível para o equipamento operar antes de ser necessária sua manutenção ou substituição. Com isso, através do histórico de temperaturas dos equipamentos é possível fazer um estudo do envelhecimento de cada componente e entender a viabilidade da continuação de suas operações ou a necessidade da troca ou manutenção dos componentes.

A usina Angra 1 não possui medidores de temperatura em locais estratégicos para o contexto do estudo de envelhecimento, dessa forma, este trabalho tem como uma de suas bases propor uma metodologia de simulação de medidores de temperatura nestas posições estratégicas necessárias, para a criação de um histórico de temperatura no interior do prédio de contenção do reator.

Angra 1 possui um sistema de computadores para monitoração de parâmetros de operação, o Sistema Integrado de Computadores de Angra (SICA) (SCHIRRU,2014), o qual gerencia parâmetros da usina através de alguns sensores tanto de radiação quanto de temperatura, localizados de forma estratégica para melhor entendimento dos estados de operação da planta.

Uma vez que a temperatura é uma medição física, pode-se partir do pressuposto de que sensores funcionais instalados num mesmo ambiente físico precisam medir temperaturas de forma semelhante. Mesmo que as medições de diferentes sensores possam não ser exatamente iguais, devido a calibração, precisam e outras razões, espera-se que as medições sejam correlacionáveis.

O Objetivo final deste trabalho é propor uma metodologia que consiste da aplicação de redes neurais artificiais, implementadas em linguagem Python, para

a simulação de medidores de temperatura nas posições, no interior do prédio de contenção, estrategicamente favoráveis ao escopo da extensão de vida útil, utilizando como base de treinamento os dados de temperatura do SICA.

No ano de 2015, já tendo o plano de extensão de vida útil como um objetivo para os próximos anos, a Eletronuclear instalou medidores de temperatura e radiação em posições estratégicas dentro do prédio de contenção do reator, na usina Angra 1, já como parte do Programa de Qualificação Ambiental de Equipamentos Elétricos (PQAEE) (IAEA, 1998, RÓNAKY,2007) . As medições feitas por esses sensores foram utilizadas em conjunto com os registros do SICA no treinamento das redes neurais, como foi proposto na metodologia desenvolvida. Os dados destes sensores da Eletronuclear também foram utilizados como base de comparação e verificação dos modelos de simulação.

Durante a etapa de pesquisa literária sobre o tema da simulação de medidores de temperatura para extensão de vida útil, foi encontrado apenas um trabalho sobre o tema, feito por Pinheiro, Victor Henrique Cabral em 2020 (PINHEIRO, 2020).

As redes neurais artificiais podem ser classificadas em algumas categorias de acordo com o objetivo de seu uso, bem como diversos outros algoritmos de aprendizado de máquina, uma dessas categorias é a regressão e é nessa categoria que se inserem as redes neurais utilizadas no escopo deste trabalho.

A regressão é um método que objetiva descrever a relação entre variáveis, com isso permitindo extrapolar as informações das quais se tem conhecimento para uma inferência de novas informações. Os algoritmos de regressão são bem conhecidos (ARCHDEACON, 1994) e o mais famoso deles, possivelmente é a regressão linear, que tem uma funcionalidade simples, uma vez que busca explicar a relação entre as variáveis com uma simples reta. Apesar de a regressão linear ser uma ferramenta extremamente poderosa para diversos casos, pela sua característica de uma saída simples ela se mostra limitada a problemas solucionáveis por essa linearidade.

De um modo geral, algoritmos regressores estão intrinsecamente ligados à dimensão do problema, sendo importante um conhecimento prévio sobre as variáveis que deseja-se correlacionar, de forma a estimar a dimensionalidade da regressão a ser feita. Nesse ponto, a grande eficiência das redes neurais regressoras se mostra presente, uma vez que um trabalho de redes neurais não necessita um conhecimento prévio sobre a possível dimensionalidade do problema.

Uma vez que as redes neurais trabalham com otimização de parâmetros através de cálculos matriciais, ela não olha para a dimensionalidade do problema, sendo assim um método muito mais preciso e eficiente de regressão.

Uma forma de entender melhor a funcionalidade de uma rede neural regressora é compará-la a uma outra categoria de algoritmos de aprendizado, a classificação. Na classificação, o algoritmo tem como objetivo separar os dados de entrada em diferentes categorias. Pode-se exemplificar um classificador com um classificador de frutas, um algoritmo especializado em receber uma fruta e devolver uma resposta dizendo se aquela fruta é uma maçã, uma banana ou uma laranja.

Em oposição aos classificadores, os regressores dão saídas numéricas, ou seja, elas não estão separadas em grupos. Para exemplificar, pode-se pensar num algoritmo para cálculo do quadrado de um número, dado um número qualquer como entrada para o algoritmo, ele deve devolver como saída o valor desse número multiplicado por ele mesmo.

Os exemplos utilizados são extremamente simples, de forma que não é necessário implementar um algoritmo para realizar tais atividades, porém no intuito de facilitar o entendimento, exemplos simples são as melhores escolhas.

Alguns trabalhos que utilizam redes neurais com a função de regressores, podem ser encontrados em (LOPES, 2011, SPECHT, 1991, DREISEITL e OHNO-MACHADO, 2002).

Para apresentar a metodologia proposta, seus fundamentos teóricos e os resultados obtidos, este trabalho foi organizado em 5 capítulos descritos sumariamente a seguir.

No presente capítulo foi feita uma introdução ao que será abordado no trabalho, apresentando e contextualizando a questão da operação de uma usina nuclear e sua extensão de vida útil.

No capítulo 2 são apresentados os fundamentos teóricos, mostrando como deve ser feito o processo de licenciamento e a solicitação para o licenciamento de extensão de vida útil, abordando principalmente o programa de qualificação ambiental de equipamentos, que fundamenta o estudo do método proposto por este trabalho. Também serão apresentados os fundamentos estatísticos, que terão uma extrema importância na compreensão das métricas utilizadas no capítulo 3. E finalmente, o capítulo será encerrado fazendo uma visão geral da história das redes neurais

artificiais, bem como algumas arquiteturas existentes, alguns problemas que surgem com essas arquiteturas e como esses problemas são solucionados.

No capítulo 3 é apresentado o problema a ser solucionado, a metodologia proposta para essa solução, sua primeira elaboração e os problemas encontrados. É apresentada também a forma como esses problemas foram solucionados.

No capítulo 4, são apresentados os resultados da implementação após a solução do problema encontrado no capítulo 3.

Finalmente, no capítulo 5 são apresentadas as conclusões a respeito dos resultados obtidos e feita uma contextualização da aplicação desses resultados no problema real da extensão de vida útil.

E por último são apresentadas as referências bibliográficas, num capítulo não numerado.

Capítulo 2

Fundamentos Teóricos

2.1 Segurança Nuclear

Atualmente a indústria nuclear em todo o mundo é referência em padrões de segurança, possuindo rígidos níveis de regulamentação para construção e operação de usinas. O objetivo final da cultura de segurança nuclear é garantir que todo o material radioativo (produtos da reação de fissão) fique confinado na planta, tanto em operação normal, quanto em caso de acidentes.

Toda usina nuclear, ainda em sua fase de projeto, deve possuir estudos sobre possíveis acidentes na etapa de operação. Sendo conhecidos os possíveis acidentes, alguns aspectos extremamente rigorosos em relação ao próprio projeto, bem como a fabricação e operação da usina são necessários para aumentar a segurança de uma instalação. Como mostrados em (CARDOZO, 2019) , pode-se citar alguns desses aspectos,

- Margens de segurança adequadas no projeto de sistemas e componentes;
- Seleção cuidadosa de materiais;
- Garantia de qualidade durante fabricação, montagem, comissionamento e operação;
- Supervisão de qualidade ao longo da vida útil da central mediante inspeções periódicas de rotina;
- Facilidade de manutenção de sistemas e componentes da planta;
- Monitoração confiável das condições de operação;
- Registro, avaliação e utilização das experiências adquiridas durante a operação com o intuito de aprimorar a segurança;

- Treinamento rigoroso e abrangente dos operadores.

Diante de toda orientação à segurança na qual as usinas nucleares são projetadas, construídas e operadas, pode-se dizer que todos os aspectos que dizem respeito a segurança da instalação e da população que vive no entorno da usina são analisados e estudados minuciosamente, para que os riscos provenientes da instalação sejam reduzidos tanto quanto possível.

As primeiras usinas nucleares foram construídas na década de 1950, desde então o setor nuclear está em constante evolução tanto no quesito de tecnologia, quanto em experiência através das operações das usinas que vão sendo construídas. Essa constante evolução, permite que seja possível a elaboração de especificações cada vez mais restritas e mais bem elaboradas, resultando em maior segurança (TODREAS, 2015).

Nas décadas de 1950 e 1960, com a construção das primeiras usinas nucleares no mundo, muito ainda era uma experiência e tudo que se sabia sobre a energia nuclear era apenas teórico, sendo necessário tudo acontecer na prática para se ter o conhecimento real de todas as coisas. Os processos de revisão de segurança eram feitos de forma que muitas vezes não se levava em conta efeitos importantes, como o envelhecimento da usina e mudanças em estados de operação. Isso acontecia possivelmente por falta de experiência prática de que esses eram, e são, fatores importantes a serem revisados.

Diante do contexto da criação de um documento que regulamente as revisões das condições das usinas nucleares, surge a Revisão Periódica de Segurança (RPS) (IAEA, 2010), como forma de padronizar as revisões que acontecem nas usinas.

A RPS proporciona uma visão geral das condições de segurança atuais da planta e de seus documentos regulatórios. Para manter as revisões num alto nível precisa-se identificar quaisquer limites de tempo de vida útil para componentes da planta, de forma a planejar-se manutenções e estimar quando deve ocorrer a próxima RPS.

Os objetivos da RPS são (IAEA, 2009):

- Determinar o tempo pelo qual a licença de operação e os documentos de segurança se mantêm válidos;
- Garantir a eficiência dos componentes que estão relacionados à segurança da

planta até a próxima RPS ou até o fim da operação da planta caso ela encerre sua operação antes de realizar-se outra RPS;

- Determinar que os padrões de segurança nacionais e internacionais estão sendo cumpridos.

A revisão de segurança deve abranger todos os aspectos de segurança da usina nuclear. Sendo assim, devem ser levados em conta todos os componentes, sistemas, estruturas em todos os seus estados de operação. É recomendado que as RPS ocorram a cada intervalo de 10 anos, intervalo considerado apropriado para avaliar mudanças tecnológicas em padrões de segurança nacional e internacional, identificar efeitos e tendências de envelhecimento de sistemas, componentes e estruturas, acumular experiência operacional e avaliar mudanças naturais, industriais e demográficas da área ao redor da planta. Dessa forma, a cada revisão de segurança realizada, deve-se garantir a operação segura de todos os componentes e sistemas pelos próximos 10 anos, até que se faça a nova RPS.

Numa usina com várias unidades, os aspectos de segurança relacionados à proteção radiológica, planos de emergência e impactos radiológicos devem ser comuns a todas as unidades. Outros aspectos como envelhecimento e condição de operação dos sistemas e componentes, podem ser tratados individualmente por unidade, de acordo com as necessidades de cada uma.

Antes das revisões de segurança começarem a ser realizadas, deve-se criar um acordo entre o órgão regulador e a organização que irá operar a RPS. Esse acordo deve abranger os padrões de segurança vigentes nacional e internacionalmente e então deve ser formulado o Documento Básico de RPS, que conterá todos os processos, referências e padrões de segurança que deverão ser aplicados às revisões.

Em usinas com tecnologias modernas e documentações atuais, onde o estudo dos fatores de segurança já foram feitos de forma mais elaborada e criteriosa, o Documento Básico da RPS torna-se mais curto e simples de ser feito. O mesmo vale para uma RPS subsequente, uma vez que na anterior, todas diretrizes já tinham sido muito bem definidas, precisando o novo documento abranger apenas as alterações que ocorreram na instalação no intervalo de operação entre uma RPS e outra.

Entrando no contexto da extensão de vida útil, tem-se a RPS como uma boa forma de iniciar esses estudos, uma vez que através das revisões, visa-se garantir todos os parâmetros de segurança no intervalo entre cada RPS. Dessa forma, essa documentação pode servir até como ferramenta de decisão sobre estender ou não a

vida útil de uma usina nuclear.

Tendo como finalidade a extensão de vida útil, algumas alterações devem ser feitas na forma como se realiza a RPS, pois deverão ser levados em conta, dentre outros efeitos, os envelhecimentos devido a radiação e temperatura dos componentes e sistemas da instalação, principalmente dos sistemas de segurança.

2.2 Extensão de Vida Útil

A extensão de vida útil consiste num licenciamento para uma operação além do tempo de operação licenciado na base de projeto. As usinas nucleares no mundo, usualmente possuem um tempo de operação licenciado de 40 anos, o que não é diferente para as usinas nucleares brasileiras.

No ano de 2014, 74 das 100 usinas que operavam nos EUA, receberam autorização de operação de longo prazo, aumentando sua vida útil de 40 para 60 anos. No mesmo ano, era esperada pelo órgão regulador norte-americano, a chegada de mais 7 solicitações de extensão de vida útil (EIA, 2014), mostrando que a extensão de vida útil das usinas nucleares é uma prática comum ao redor do mundo.

Chegando no final do tempo de vida útil de uma usina, surge a preocupação com o descomissionamento (WNA, 2019), que é um processo de altos custos financeiros e de mobilização de pessoal. Dessa forma surge a alternativa da extensão de vida útil, onde se solicita ao órgão regulador, uma licença para estender o tempo de operação da usina por mais 20 anos.

Devido ao alto custo do descomissionamento e da importância na produção de energia elétrica e no seu custo de venda, a extensão de vida útil de 40 para 60 anos, se mostra financeiramente viável se os principais componentes da usina estiverem em ótimos estados e as mudanças que devem ser feitas para essa extensão de operação não tenham altos impactos estruturais e operacionais estabelecidos no licenciamento.

No ano de 1982, a NRC (*Nuclear Regulatory Commission*), órgão norte-americano de regulação das tecnologias nucleares, estabeleceu um programa para estudar e avaliar os processos de envelhecimento em usinas nucleares e com os resultados desses estudos constatou-se que o envelhecimento de estruturas, sistemas e componentes não apresenta nenhum problema técnico que não possa ser gerenciável, dessa forma não havendo um impeditivo para a implementação de programas de

extensão de vida útil.

Na década seguinte, foi publicado um documento, o 10 CFR parte 54 (NRC, 1954), com os requisitos para a renovação da licença além da vida útil de base de projeto das usinas nucleares. Esse documento tem por objetivo concentrar-se nos efeitos adversos do envelhecimento das estruturas, sistemas e componentes relevantes à segurança e garantir que continuem a operar da forma desejada nos anos de operação de longo prazo.

A renovação da licença só é concedida pela NRC, caso todos os requisitos da 10 CFR parte 54 estejam sendo cumpridos e esse documento de solicitação da renovação de licença, deve ser enviado à NRC com um prazo de pelo menos 5 anos antes da expiração da licença atual.

No Brasil, a renovação de licença das usinas nucleares é feita pela CNEN , através na norma NE 1.04 (CNEN, 1984). Dessa forma existe uma norma nacional satisfatória que trata do licenciamento de usinas nucleares e garante a operação segura das usinas. Porém como Angra 1 é a primeira usina nuclear brasileira em processo de extensão de vida útil, não sabe-se até que ponto a norma NE 1.04 da CNEN garante a segurança em operação de longo prazo. Devido a isso acredita-se que utilizar a experiência do regulamento 10 CFR parte 54 da NRC em associação com a norma NE 1.04 seja uma alternativa para suprir a não existência de uma experiência anterior de extensão de vida útil no Brasil (ELLER, 2018).

Para a extensão de vida útil, deve-se solicitar à CNEN, a renovação da licença até 5 anos antes do fim do período original para operação. O documento de solicitação deve garantir que as condições das estruturas, sistemas e componentes serão gerenciadas de forma que sejam mantidas constantes com a base de licenciamento original, durante o período de operação de longo prazo.

A solicitação de extensão de vida útil deve rever e validar todos os programas que tenham relação direta com o processo de envelhecimento das estruturas, sistemas e componentes da usina.

A usina Angra 1, por ter seu prazo de operação até o ano de 2024 começou estudos para extensão de vida útil em 2015, porém desde 2009, quando fez a troca do gerador de vapor, vem investindo nesse objetivo, onde cerca de R\$ 1 bilhão de reais estão sendo destinados para estender a vida útil por mais 20 anos. Cabe ressaltar que com esse investimento o país estará garantindo 640 MW de energia

elétrica por mais 20 anos, sendo a extensão de vida útil uma forma barata de obtenção de energia (DE SOUZA, 2019).

Para se fazer uma avaliação eficaz das condições de operação de longo prazo, deve-se levar em conta o envelhecimento dos equipamentos, estruturas e sistemas dentro da usina. O processo de envelhecimento consiste na degradação devido ao tempo de operação tanto em condições normais, quando em condições de transientes. Como o período de operação de uma usina nuclear é longo, é comum uma gradual alteração nas propriedades dos materiais. Apesar de essas mudanças nem sempre serem degradantes, elas podem afetar a capacidade de operação completa de alguns equipamentos.

Todos os materiais dentro de uma usina nuclear podem sofrer envelhecimento, mas deve-se dar uma atenção especial àqueles que trabalham em maiores temperaturas e recebem maiores doses de radiação, devido a essas duas condições serem agravantes ao processo de envelhecimento.

O envelhecimento pode afetar desde a eficiência na geração de energia elétrica, podendo levar a um prejuízo econômico para a operadora da usina até as condições de segurança, sendo assim um fator de extrema importância a ser analisado tanto nas RPS, quando num programa de extensão de vida útil (MILAN, 1987).

Dentre os programas que tratam da questão do envelhecimento nos equipamentos da usina, se encontra o Programa de Qualificação Ambiental de Equipamentos Elétricos (PQAEE) (IAEA, 1998, RÓNÁKY, 2007), onde está inserida a base deste trabalho.

2.2.1 Programa de Qualificação de Equipamentos

O processo de qualificação de equipamentos começa durante a fase de desenvolvimento da usina e continua até o final de sua vida útil. Para que a qualificação seja feita de forma eficiente, deve-se trabalhar com processos simples e cuidadosos no lugar de processos complexos de qualificação, seguindo todas as regulamentações do órgão regulador.

A qualificação ambiental de equipamentos segue 3 categorias (CARDOZO, 2019):

- Equipamentos relacionados a segurança, que devem permanecer em funcionamento seguindo as bases de projeto para assegurar:
 - Integridade do Sistema de Refrigeração do Reator;

- Capacidade de desligar e manter o reator em condições seguras de desligamento;
 - Capacidade de prevenir e/ou mitigar as consequências de acidentes.
- Equipamentos elétricos não relacionados a segurança, que consistem em equipamentos cuja falha pode provocar um funcionamento não satisfatório dos equipamentos diretamente relacionados à segurança.
 - Equipamentos de monitoração pós-acidente:
 - Qualificação dos equipamentos de instrumentação em condições severas de operação em casos de acidente.

Componentes elétricos de instrumentação e controle exigem uma qualificação individual de cada componente para garantir disponibilidade, operação segura e confiabilidade durante todo seu tempo de operação. A qualificação visa determinar parâmetros de operação que estarão dentro das condições ambientais às quais os equipamentos irão operar, dessa forma garantindo que a qualificação permaneça válida.

A questão da qualificação visando as condições do ambiente de operação estão ligadas tanto ao fato de que os equipamentos precisam aguentar as condições locais da instalação, quanto ao fato de que o maior causador de falhas de causa comum em equipamentos é o ambiente onde eles se encontram instalados. Alguns aspectos relacionados ao ambiente que podem influenciar na falha dos equipamentos são temperatura, radiação, umidade, pressão e eles estão ligados diretamente a localização de operação, assim como a condições de acidente, onde esses parâmetros podem se alterar de forma drástica.

Assim como alguns fatores estão ligados a condições ambientais, há parâmetros ligados à operação da usina, como fluxos, vibrações próprias e mesmo temperatura e radiação. Apesar de a abordagem da qualificação ser ligeiramente diferente para cada tipo de parâmetro dependendo da origem da causa de sua falha, todos devem ser levados em conta na qualificação inicial de equipamentos.

A qualificação dos equipamentos que realizam funções de segurança, instalados na usina, deve garantir que esses equipamentos irão continuar operando independente do estado da planta, para que sejam capazes de mitigar os efeitos da liberação radioativa em casos de acidente e essa qualificação deve permanecer válida durante toda a vida útil desses equipamentos.

Com o programa de qualificação dos equipamentos instalado, deve ser iniciado o processo de manutenção da qualificação, que tem por objetivo garantir que os equipamentos, sistemas e estruturas mantenham sua qualificação ao longo de sua vida útil. A base desse processo de garantia é a vigilância e a inspeção, através de testes em operação, monitoramento e diagnóstico de mudanças de estados.

Um programa de qualificação bem elaborado dirá quando e com que frequência devem ser realizadas manutenções e/ou trocas de equipamentos que estejam chegando perto do fim de sua vida útil e essa substituição deve ser feita de maneira preventiva, para garantir que não atinjam a falha, prejudicando qualquer sistema de operação, garantindo que tudo continue operando de forma segura.

O processo de monitoramento deve ser elaborado de forma a monitorar continuamente e de forma não destrutiva o processo de envelhecimento de estruturas, sistemas e componentes que desempenham função de segurança, de forma a ter um controle de que esse processo de envelhecimento ocorre de acordo com o previsto no programa de qualificação.

O diagnóstico das condições do sistema faz com que seja elaborada uma avaliação rápida dos estados de operação e condições ambientais de cada equipamento, estrutura e sistema e podem indicar o surgimento de falhas e a necessidade de manutenção corretiva (IAEA, 1998, RÓNAKY, 2007).

O programa de qualificação ambiental visa garantir que as estruturas, sistemas e componentes da usina tenham condições de operar em situações normais e de acidente, minimizando a possibilidade de acidentes, mitigando as consequências de possíveis acidentes e impedindo a liberação de material radioativo para o meio ambiente. Essas funções serão cumpridas, garantindo que os componentes tenham condições de manter a integridade das barreiras de segurança, que impedem que material radioativo escape, mantendo condições seguras para um desligamento de emergência do reator e garantindo que o reator terá sempre um sistema de resfriamento, para que mesmo que esteja desligado o núcleo não possa derreter.

Um programa de qualificação ambiental é realizado mediante três fases:

- **Fase de Planejamento:** Onde são definidas todas as variáveis das condições de operação, tanto em condições normais quanto de acidente, da planta.
- **Fase de Qualificação:** Fase onde todos os testes nos equipamentos é realizada utilizando todas as condições estabelecidas na fase de planejamento.

- **Preservação da Qualificação:** Uma vez estabelecida a qualificação dos equipamentos, essa qualificação deve ser preservada, garantindo que todos os componentes continuem operando de forma eficiente.

2.3 Modelo de Envelhecimento de Arrhenius

O programa de qualificação ambiental necessita de estudos sobre a qualificação dos equipamentos presentes na usina, para que este possa ser elaborado e o licenciamento de longo prazo possa ser solicitado ao órgão regulador. A vida qualificada dos equipamentos no interior do prédio de contenção do reator está diretamente ligada ao seu envelhecimento, que nada mais é do que um processo de degradação, que ocorre devido a exposição a temperaturas mais elevadas.

O modelo térmico de envelhecimento de Arrhenius (DAVID, 1992) é comumente utilizado na indústria como um todo para estudos e estimativas de envelhecimento e vida útil de equipamentos. Sua formulação matemática é relativamente simples, dependendo de alguns parâmetros físicos dos equipamentos, das temperaturas a qual foram expostos e do tempo ao qual foram expostos a essa temperatura.

Uma forma de representar a formulação matemática em função do cálculo da vida útil dos equipamentos é apresentada em (PINHEIRO, 2020), representada na equação 2.1.

$$\frac{t_s}{t_a} = e^{\left(\frac{\Phi}{k}\right)\left[\frac{1}{T_s} - \frac{1}{T_a}\right]} \quad (2.1)$$

Onde:

- Φ : Energia de Ativação (eV);
- k : Constante de Boltzman ($8,617 \cdot 10^{-5} eV/K$);
- t_a : Tempo de envelhecimento acelerado (h);
- t_s : Tempo de operação (h);
- T_a : Temperatura de envelhecimento (K);
- T_s : Temperatura de operação (K).

O envelhecimento é um processo químico de degradação que ocorre nos equipamentos, modificando sua estrutura molecular. Todo processo químico ocorre

mediante uma energia de ativação, abaixo da qual a reação química não ocorre. Na equação 2.1, Φ representa essa energia de ativação e usualmente essa energia para ativar o processo químico é fornecida na forma de calor, quando o equipamento opera a temperaturas mais elevadas.

Todos os equipamentos em numa usina nuclear são previamente qualificados antes de serem instalados, dessa forma, a temperatura de operação (T_s) e o tempo de operação (t_s) são conhecidos, porém esse comportamento teórico se modifica quando na prática os equipamentos são expostos a temperaturas acima daquela estipulada em sua qualificação, o que é comum ocorrer em usinas nucleares devido aos transientes que ocorrem nos modos de operação.

Com essa exposição a temperaturas mais elevadas (T_a), naturalmente o tempo ao qual o equipamento foi qualificado para operar é reduzido para o que é representado pelo tempo de envelhecimento acelerado (t_a) na modelagem de Arrhenius.

A partir do entendimento da importância da temperatura no tempo em que um equipamento se mantém qualificado para operar é possível entender a relevância do estudo realizado neste trabalho, na inferência de um histórico de temperaturas de operação no interior do prédio de contenção da usina nuclear.

2.4 Fundamentação Estatística

As atividades em *Machine Learning* (MICHALSKI e TECUCI, 1994) nada mais são do que uma combinação de conhecimento em ciências da computação, álgebra linear, estatística e dados. Apesar de todas as áreas de conhecimento serem igualmente importantes, é necessário passar por alguns fundamentos estatísticos que serão essenciais na análise e compreensão dos dados que foram utilizados neste trabalho.

A estatística é definida como uma ciência matemática, que se preocupa com a coleta, análise e interpretação dos dados, utilizando para isso diversos métodos e modelos matemáticos, que se fundamentam em teorias probabilísticas. Acaba sendo curioso fazer uma correlação de que a estatística, apesar de ser uma ciência matemática, não é uma ciência exata, pois trabalha com estimativas e probabilidades de forma a entender comportamentos.

Associada à ciência da computação e à álgebra linear, cria-se uma ferramenta

poderosa na construção de modelos preditivos, capazes de com muita precisão estimar possíveis eventos futuros a partir de dados coletados.

Serão abordados os seguintes conceitos estatísticos,

- Distribuição Gaussiana
- Correlação de Pearson
- Correlação de Spearman

2.4.1 Distribuição Gaussiana

A distribuição Gaussiana (MARSAGLIA, 2004), mais popularmente chamada de distribuição normal, está entre as distribuições mais importantes e conhecidas na estatística. Seus primeiros aparecimentos aconteceram no ano de 1733, quando Abraham de Moivre estudava o fatorial, mas foi apenas em 1809, nas mãos de Carl Friedrich Gauss, que a distribuição começou a ser modelada da forma como é conhecida atualmente, quando ele assimilou os erros da observação astronômica à curva da distribuição, ficando então conhecida como Gaussiana, em homenagem ao matemático, físico e astrônomo alemão.

A distribuição normal (MARSAGLIA, 2004) em sua essência é uma forma de curva, mas existe uma infinidade de curvas com o mesmo formato, porém dimensões diferentes. O que torna a Gaussiana realmente importante é o conceito da distribuição normal padrão, uma distribuição com a forma da Gaussiana onde a média $\mu = 0$ e o desvio padrão $\sigma = 1$.

A distribuição normal padrão tem sua escala de eixo X denotada pelo *escore* $- z$, uma escala de medida de posição que indica a quantidade de desvios padrão que uma variável se encontra distante da média. A grande importância da normal padrão é a possibilidade de transformar qualquer distribuição normal com variável x em uma distribuição normal padrão através da relação apresentada na equação 2.2

$$z = \frac{x - \mu}{\sigma} \tag{2.2}$$

onde,

- x é a variável aleatória inicial;
- μ é a média da distribuição;
- σ é o desvio padrão da distribuição

A distribuição normal padrão possui algumas características importantes de serem mencionadas,

- A área acumulada é próxima de 0 para $score - z \leq -3.49$;
- A área acumulada para $z = 0$ é 0.5;
- A área acumulada é próxima de 1 para $score - z \geq 3.49$;

Como a área acumulada representa a probabilidade de um evento, convertendo-se uma distribuição normal comum em uma normal padrão, processo chamado de padronização da distribuição normal, rapidamente pode-se observar essas propriedades e fazer inferência de uma informação. Além dessas propriedades, tem-se tabelados todos os valores de probabilidade para o $score - z$, o que se mostra muito eficiente em termos de cálculo, uma vez que o cálculo algébrico da probabilidade não é possível de forma analítica, como será apresentado na equação 2.4.

A função de densidade da distribuição normal, chamada de função de Gauss, é representada por $\varphi(t)$,

$$\varphi(t) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2} \cdot t^2} \quad (2.3)$$

A função de Gauss é contínua e infinitamente diferenciável, por ser uma composição de exponencial. Sua integral em t , representa a probabilidade acumulada de um evento, essa integral é apresentada na equação 2.4,

$$\Phi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t e^{-\frac{1}{2} \cdot t^2} dt \quad (2.4)$$

Dadas as formulações da distribuição normal, pode-se aprofundar alguns conceitos que foram de crucial importância na análise dos dados como será apresentado no capítulo 3.

Momento

Em estatística, o momento de uma função é uma medida quantitativa relacionada ao formato de sua curva. Em uma distribuição de probabilidade, o primeiro momento representa o valor esperado da variável aleatória, em uma distribuição

normal pode-se dizer que representa a média μ , o segundo momento, representa a variância, o terceiro momento, quando padronizado, representa a *skewness*, medida de assimetria, e o quarto momento representa a *kurtosis*, também quando padronizado.

Definindo $f(x)$ como a função de distribuição, o n-ésimo momento desta pode ser calculado para distribuições contínuas, por:

$$\mu_n = \int_{-\infty}^{+\infty} x^n \cdot f(x) dx \quad (2.5)$$

e para uma distribuição discreta, por:

$$\mu_n = \sum x_i^n \cdot f(x_i) \quad (2.6)$$

Essa breve passagem pelo conceito de momento é importante no entendimento da definição da *kurtosis* e da *skewness*.

Skewness

A *skewness* é uma medida de assimetria de uma distribuição, indicando o quanto ela é simétrica em relação à normal padrão. É definida como a padronização do terceiro momento,

$$\gamma_1 = \frac{\mu_3}{\sigma^3} \quad (2.7)$$

Uma medida de *skewness* igual a 0 representa uma distribuição normal perfeita. Valores negativos indicam uma assimetria para a direita, criando uma cauda alongada para a esquerda. Valores positivos representam uma distribuição com concentração de valores abaixo da média, ou seja, cauda alongada para a direita.

Kurtosis

A *kurtosis* é uma medição do achatamento da curva em relação a distribuição normal, é definida como o a padronização do quarto momento,

$$\gamma_2 = \frac{\mu_4}{\sigma^4} - 3 \quad (2.8)$$

Valores de *kurtosis* iguais a 0 indicam igualdade com a distribuição normal, enquanto valores negativos indicam uma curva achatada e valores positivos indicam um alongamento da curva, tornando-a mais alta que a gaussiana.

2.4.2 Medidas de Correlação

Medidas de correlação (ARCHDEACON, 1994) são métodos estatísticos de relacionamento entre variáveis. Seu objetivo é entender como uma variável se comporta em relação a variação de uma outra variável e se existe ou não uma relação entre elas. Os coeficientes de correlação não indicam causalidade, ou seja, não indicam que uma variação em uma variável é a causa da variação da outra, mas indicam que quando uma varia, outra também varia em determinado sentido.

Podem existir três tipos de correlação, positiva, indicando que ambas as variáveis se alteram no mesmo sentido, negativa, quando se alteram em sentidos opostos, ou neutra, quando não se pode observar nenhuma relação entre ambas.

Correlação de Pearson

O coeficiente de correlação de Pearson (BENESTY *et al.*, 2008) é definido como a métrica de correlação linear entre dois *datasets*. O cálculo do valor-p é feito assumindo que ambos os *datasets* possuem uma distribuição normal. Possivelmente é o coeficiente de correlação mais conhecido na estatística, mas possui a característica de que para seu valor ser representativo, os *datasets* precisam seguir distribuições normais. Esse coeficiente é calculado em função da variância *var* e da covariância *cov* da distribuição, como segue,

$$\rho = \frac{cov(X, Y)}{\sqrt{var(X).var(Y)}} \quad (2.9)$$

Correlação de Spearman

Definida como coeficiente de correlação de postos de Spearman (SPEARMAN, 1904), é uma medida não paramétrica da monotonicidade da relação entre dois *datasets*. Diferente da correlação de Pearson, a correlação de Spearman não assume que os *datasets* possuam distribuição normal. Seu cálculo é feito através da conversão

da distribuição em postos rg_X e rg_Y e então é aplicada a correlação de Pearson nesses postos, como segue,

$$r_s = \rho_{rg_X,rg_Y} = \frac{cov(rg_X,rg_Y)}{\sqrt{var(rg_X).var(rg_Y)}} \quad (2.10)$$

Ambas correlações retornam valores entre -1 e $+1$, valores mais próximos dos extremos indicam forte correlação negativa e positiva, respectivamente. Valores mais próximos a 0 indicam baixa correlação.

2.5 Redes Neurais Artificiais

2.5.1 História das Redes Neurais Artificiais

As Redes neurais artificiais são um sub conjunto de algoritmos de aprendizado de máquina. Os primeiros passos no desenvolvimento das redes neurais artificiais foram dados em 1943, quando Warren Culloch e Walter Pitts, tentando entender o funcionamento do cérebro humano (a rede neural biológica), modelaram uma rede neural simples, usando circuitos elétricos (MCCULLOCH e PITTS, 1943).

Seis anos depois, em 1949, Donald Hebb escreveu que os caminhos neurais são fortalecidos a cada vez que são usados, o que explica o aprendizado humano, que consiste num processo de repetição e/ou aplicação de um conhecimento. A partir da obra de Donald Hebb junto com a evolução da capacidade computacional por volta nos anos 1950, começaram a surgir as tentativas de reproduzir uma rede neural de forma computacional (HEBB, 1949).

Chegando ao final dos anos 1950, Frank Rosenblatt começou a trabalhar no Perceptron, que foi a primeira rede neural artificial criada e que ainda pode ser usada até hoje (ROSENBLATT, 1957). No ano de 1969, no livro *Perceptrons* (MINSKY e PAPERT, 1969), de Marvin Minsky e Seymour Papert, foi comprovado que o *perceptron* tem uma capacidade muito limitada. Mas não pode-se descartar a importância do *perceptron* para os modelos mais avançados de redes neurais artificiais, pois seu conceito simples serviu de base para diversas arquiteturas mais complexas que vieram a seguir.

A especulação que se seguiu aos primeiros passos com inteligência artificial, foi de que ela iria em breve substituir os humanos, que robôs pensantes seriam uma realidade em poucos anos. Cerca de 4 a 5 décadas depois essa discussão ainda prevalece.

Ao longo do ano de 1981, as pesquisas e desenvolvimentos no ramo de inteligência artificial diminuíram consideravelmente e esse evento ficou conhecido como Inverno da IA.

Em 1987, aconteceu a primeira conferência internacional sobre redes neurais do *Institute of Electrical and Electronic Engineer's* (IEEE) , atraindo mais de 1800 participantes. Seguindo a essa conferência, os primeiros algoritmos de aprendizado profundo foram desenvolvidos, porém seus tempos de treinamento estavam sendo medidos em dias e suas aplicações sendo impraticáveis em problemas do mundo real. E nesse ponto chegou-se a conclusão de que o desempenho da IA começava a depender de hardwares mais potentes, algo que não estava disponível tecnologicamente na época.

O salto exponencial da Inteligência artificial começou com a chegada dos anos 2000 e com o avanço também exponencial das tecnologias de hardware, possibilitando a retomada dos estudos e desenvolvimento de algoritmos, porque passou-se a ter disponível máquinas capazes de trabalharem com esses algoritmos.

Em 2012 os primeiros algoritmos a atingirem um desempenho em nível humano em algumas tarefas começaram a surgir. Em 2014, a DeepMind, uma startup de IA foi comprada pelo Google por 400 milhões de euros. Em 2015, o Facebook coloca em operação a DeepFace, uma rede neural capaz de identificar usuários da rede Facebook de forma automática, apenas pela imagem do rosto do usuário, essa rede treinava cerca de 120 milhões de parâmetros na época.

Em 2017, mais do que consolidados, os algoritmos de Deep Learning, já sendo utilizados em inúmeras aplicações, passam a ser considerados como a principal e mais promissora tecnologia de aprendizado de máquina, de inteligência artificial e de ciência de dados.

Apesar de as redes neurais modernas serem capazes de realizar tarefas que antes apenas seres humanos eram capazes, como identificar gatos e cachorros em imagens ou identificar sons, por exemplo, a proposta das redes neurais artificiais deixou de ser tentar reproduzir o pensamento humano e sim ser utilizada para aprender coisas que seriam muito difíceis, ou até mesmo impossíveis, para os seres humanos devido a característica da enorme quantidade de dados que um computador pode processar e um ser humano, não.

2.5.2 O Neurônio Biológico

O neurônio é a unidade celular do cérebro humano, sendo responsável pela transmissão de sinais nervosos. Essas células são especializadas no recebimento, processamento e encaminhamento desses sinais. É constituída por três partes principais, o corpo, onde acontece o “processamento” da informação e onde estão contidos os dendritos, que se ligam a outros neurônios e por onde a informação primeira chega, o axônio que funciona como uma ligação aos nervos terminais, que são a última parte do neurônio, onde a informação é passada adiante.

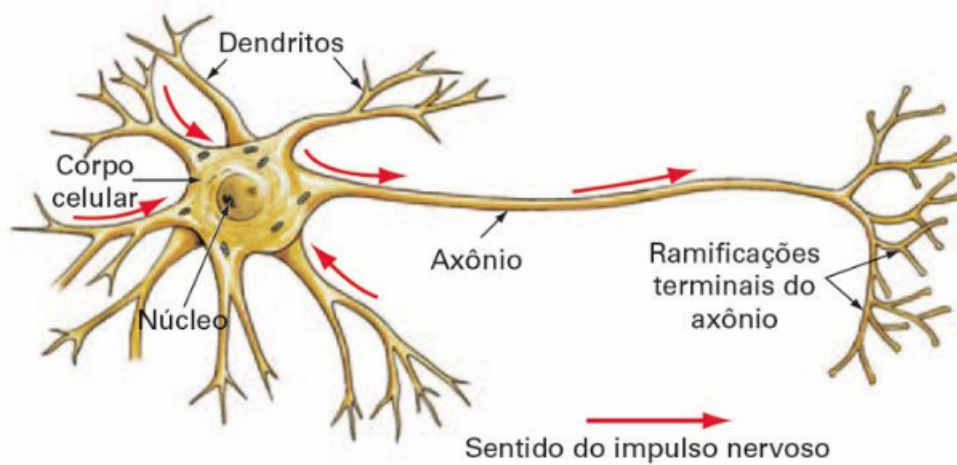


Figura 2.1: Representação do Neurônio Biológico (DSA, 2018).

O cérebro é constituído por bilhões de neurônios interligados, formando uma rede neural. Todo o trabalho começa quando algum dos neurônios sofre um estímulo e através disso passa esse estímulo para os próximos neurônios por seus nervos terminais, os neurônios seguintes recebem esse estímulo por seus dendritos, processo conhecido como sinapse. No corpo celular existe um certo limite de disparo (*threshold*), onde se o sinal recebido for superior a esse valor, ele passa para os próximos neurônios da rede, caso contrário, ele é retido naquela célula.

A existência do *threshold* permite que cada célula neural se adapte, gerando o que é conhecido como conhecimento e memória. No cérebro, os sinais que excitam os neurônios são elétricos e suas passagens entre cada célula da rede é química, através de um neurotransmissor chamado serotonina.

Entendendo o básico do funcionamento do cérebro humano e inspirado por ele, foram desenvolvidos os neurônios matemáticos, que se tornariam a base da

inteligência artificial.

2.5.3 O Neurônio Matemático

O neurônio matemático (JAIN, 1996) foi modelado como uma célula matemática onde vários *inputs* são recebidos e uma soma ponderada é calculada, para que o valor resultante seja passado adiante. O que seriam os dendritos de uma célula biológica, são formados pelos pesos matemáticos, números reais, pelos quais os inputs são multiplicados antes de chegar ao corpo. O corpo é formado por dois módulos, um primeiro onde é feito um somatório de todos os *inputs* multiplicados pelos pesos e um *bias*. O valor desse somatório passa então para uma função de ativação, que constitui o segundo módulo do corpo e então segue para o *output*, que faz o papel dos nervos terminais, passando o valor numérico para os próximos neurônios.

Por se tratar de um modelo matemático, toda a estrutura é feita utilizando cálculos numéricos. Os *inputs* mencionados, são números representados pela letra X e representam uma informação que está sendo passada para a rede. Os pesos, são representados por W e finalmente o valor que é recebido pelo corpo do neurônio após a multiplicação é representado por Y .

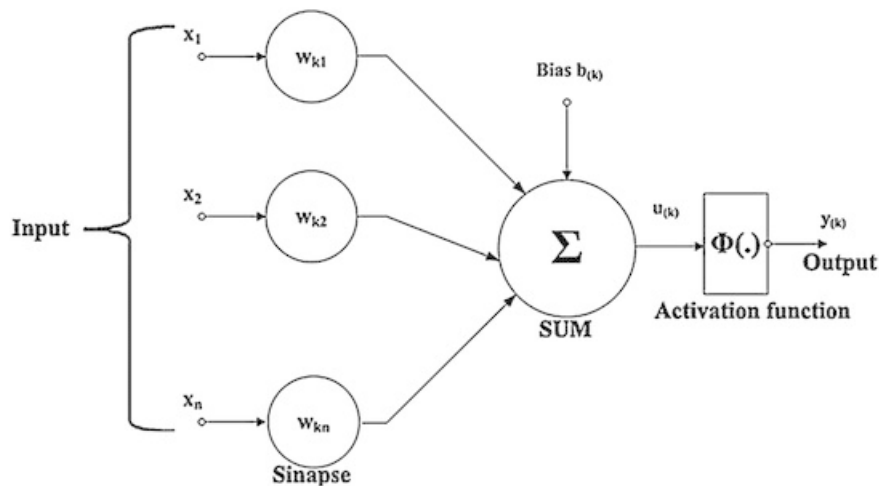


Figura 2.2: Representação do Neurônio Matemático (DSA, 2018).

Por se tratar de uma rede de conexões, a multiplicação de X por W que resulta nos valores dos Y , que o corpo do receberá para fazer o somatório, é feita através de um produto matricial. Os valores de Y são então somados pelo corpo e enviados a função de ativação. Todas as funções de ativação utilizadas em redes neurais na

literatura, tem como característica possuir um limite inferior e superior, fazendo com que o valor recebido pelo somatório seja suavizado para dentro desses limites, fazendo com que não haja um valor extremamente alto sendo passado entre os neurônios, o que poderia facilmente, dada a característica multiplicativa da rede, levar as informações trocadas entre células a ordem de grandezas superiores a capacidade dos computadores processarem.

$$\mathbf{X}\mathbf{w} = \mathbf{y}$$

$$\begin{bmatrix} 1 & x_{11} & \dots & x_{1d} \\ 1 & x_{21} & \dots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \dots & x_{nd} \end{bmatrix} \times \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Figura 2.3: Multiplicação Matricial que ocorre entre as entradas X e os pesos W (DSA, 2018).

Com a criação do neurônio matemático, diversas arquiteturas foram surgindo a partir de diferentes combinações entre neurônios. A primeira e mais simples arquitetura a se mostrar funcional, foi o perceptron.

2.5.4 O Perceptron

Como já mencionado no item 2.5.1 o primeiro perceptron foi desenvolvido no ano de 1958 por Frank Rosenblatt (ROSENBLATT, 1957), sendo a menor unidade de uma rede neural, pois ele conta com apenas um neurônio. Atualmente existem outros tipos de unidades celulares de redes neurais, mas o perceptron é o mais simples e fácil de ser compreendido.

Ele pode receber várias entradas X_i , onde cada uma dessas entradas é multiplicada por um peso W_i , cujo produto então chega na célula principal do neurônio, onde ele fará o somatório das entradas e através de uma função de ativação colocará esse valor do somatório em uma escala à qual a função de ativação está definida, usualmente $(0, 1)$ ou $(-1, 1)$, então esse valor após a função de ativação será comparado com um *threshold* definido para o problema em questão e caso seja superior, o perceptron emite um *output* igual a 1. Caso o valor seja menor que o *threshold*, o *output* será igual a 0. Usualmente seleciona-se as saídas iguais a 1 para uma continuidade do sinal, e a saída igual a 0 para o bloqueio, tendo assim o

perceptron a característica de funcionar como um filtro.

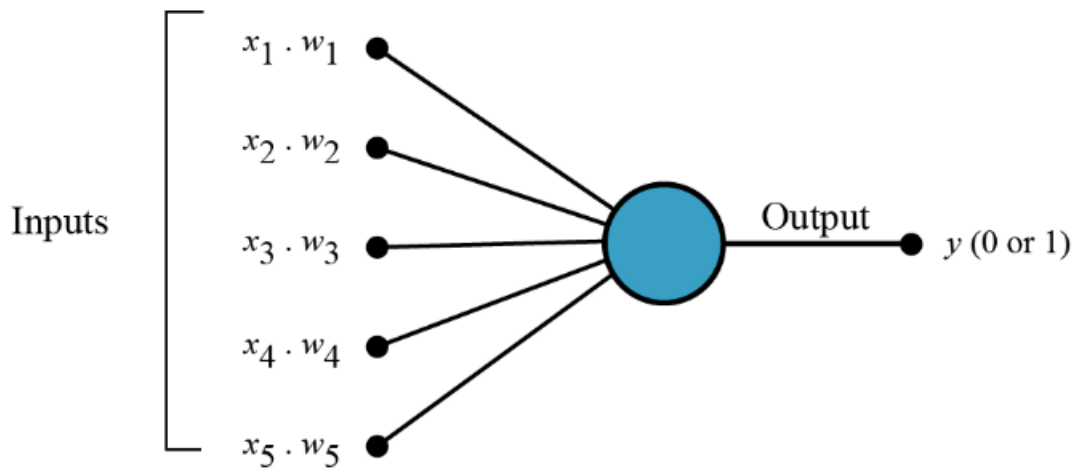


Figura 2.4: Representação do Perceptron.

Na Figura 2.4 pode-se notar que a representação do perceptron é idêntica a representação do neurônio matemático, mostrado na figura 2.2, uma vez que ele nada mais é do que uma rede de um só neurônio.

Pode-se utilizar um exemplo simples para entender o funcionamento de um perceptron na prática. Supondo uma conferência de inteligência artificial acontecendo na semana seguinte. Um estudante da área de tecnologia decide utilizar um perceptron para decidir se irá ou não ao evento, pensa que 3 fatores são decisivos para ir ou não à conferência:

- O valor da entrada;
- Se seus amigos irão acompanhá-lo;
- A facilidade de transporte até o local.

Com isso têm-se as três entradas, caso o valor da entrada esteja dentro do orçamento, essa variável passará um valor 1 para o perceptron, caso contrário, 0. Se os amigos do estudante forem acompanhá-lo, 1 será passado, caso contrário, 0. O mesmo para a facilidade de transporte, se o local da conferência for de fácil acesso, o perceptron recebe 1, caso não seja, recebe 0.

Com essa modelagem, o estudante tem os valores de X_i de entrada e precisa definir os pesos W_i . Ele acredita que o valor do ingresso é de extrema importância, então decide que o peso dessa variável será 6, mas ele não se importa tanto em ir sozinho ou mesmo que o local seja de difícil acesso, uma vez que ele está interessado no evento, então decide que os pesos para essas duas variáveis seja 2 para cada uma.

A única coisa que falta decidir é o *threshold*, que o estudante decide colocar como 5. Para o problema modelado, o Perceptron dará uma saída igual a 1, sempre que o valor da entrada for acessível e 0 em qualquer outra situação, indicando que o estudante só deverá ir à conferência, caso ele possa pagar pelos ingressos de entrada.

Uma tomada de decisão humana desse tipo, não precisaria ser modelada através de uma rede neural, o exemplo dado foi apenas uma forma lúdica de exemplificar o funcionamento simples do perceptron.

A limitação do perceptron descoberta por Marvin Minsky e Seymour Papert em 1969 foi em relação a sua capacidade de resolver apenas problemas simples e lineares, o que limita sua utilização, uma vez que no mundo real, a quantidade de problemas lineares simples que precisam de um computador para serem resolvidos são praticamente nulas. Sendo assim, novas arquiteturas precisavam ser desenvolvidas e é nesse ponto onde entram as redes neurais profundas, que são combinações de perceptrons.

2.5.5 Redes Neurais Profundas

Para compreender as redes neurais profundas, é necessário entender que toda rede neural possui pelo menos uma entrada e pelo menos uma saída, sendo assim o perceptron, uma rede neural. Mas uma rede neural com dezenas de entradas e dezenas de saídas, apenas, não pode ser classificada como profunda, uma vez que não possui camadas ocultas, ou internas (GOODFELLOW, 2016).

As camadas ocultas de uma rede neural são intermediárias entre as entradas e as saídas, onde ocorrem a maior parte dos cálculos matemáticos responsáveis pelo aprendizado das redes neurais. Uma rede neural pode ter apenas uma camada de entrada e uma camada de saída, mas pode ter inúmeras camadas ocultas, sendo a imaginação, e o poder computacional, é claro, o limite. Por se tratarem de combinações de camadas de perceptrons, essas redes receberam o nome de Perceptrons de Multicamadas ou MLP (*MultiLayers Perceptrons*).

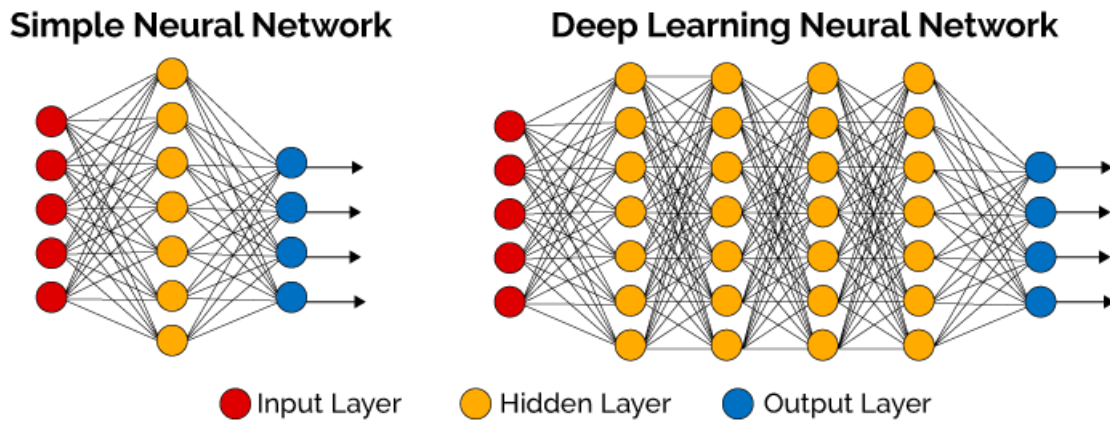


Figura 2.5: Representação de uma Rede Neural Simples, à esquerda, e de uma rede neural profunda, à direita (DSA, 2018).

Apesar de o termo MLP ser genérico para todas as redes neurais multicamadas, um termo que surgiu e é mais utilizado para se referenciar as redes neurais profundas, com muitas camadas internas, é o *Deep Neural Network* (DNN), dessa forma a nomenclatura utilizada nesse trabalho será a de DNN, por ser mais representativa às redes utilizadas aqui.

As DNNs são muito utilizadas em problemas de treinamento supervisionado, onde uma entrada e uma saída esperada são passadas para a rede. A rede utiliza então a entrada, passa essa informação adiante através da multiplicação matricial dos pesos entre neurônios, processo conhecido como *feedforward* (HORNIK e STINCHOMBE, 1992) e dá uma saída. Essa saída é então comparada com a saída esperada e o erro é calculado através de alguma função de erro, como o erro quadrático médio MSE (*Mean Squared Error*), por exemplo.

O erro encontrado ao final do processo de *feedforward* é então utilizado no *backpropagation* (WERBOS, 1990), algoritmo que ajusta os pesos de cada camada, invertendo o fluxo de passagem da informação, indo da camada de saída, para a camada de entrada (por isso o nome *backpropagation*). Chegando novamente na camada de entrada com os pesos corrigidos, as entradas são novamente passadas, o processo do *feedforward* ocorre novamente e tem-se uma nova saída, onde espera-se que o erro encontrado seja menor que o anterior e então repete-se o processo, até que o erro encontrado na camada de saída seja menor do que o erro desejado para o problema.

Descida do Gradiente

Grande parte das tarefas em *Machine Learning* consistem da otimização de parâmetros. Em redes neurais artificiais não é diferente, uma vez que é necessário fazer uma otimização dos pesos e vieses em busca da melhor função de custo possível. Um dos algoritmos mais utilizados para essa otimização é o algoritmo de descida do gradiente CAUCHY, 1847).

O gradiente é um conceito matemático de cálculo vetorial, que indica a direção de um deslocamento, no qual dada variável terá o maior incremento (ou decremento) em seu valor. A ideia da descida do gradiente remete a esse conceito, com o objetivo de causar um decremento na valor das variáveis.

Pode-se exemplificar o algoritmo com o gráfico para uma variável, apresentado na figura 2.6,

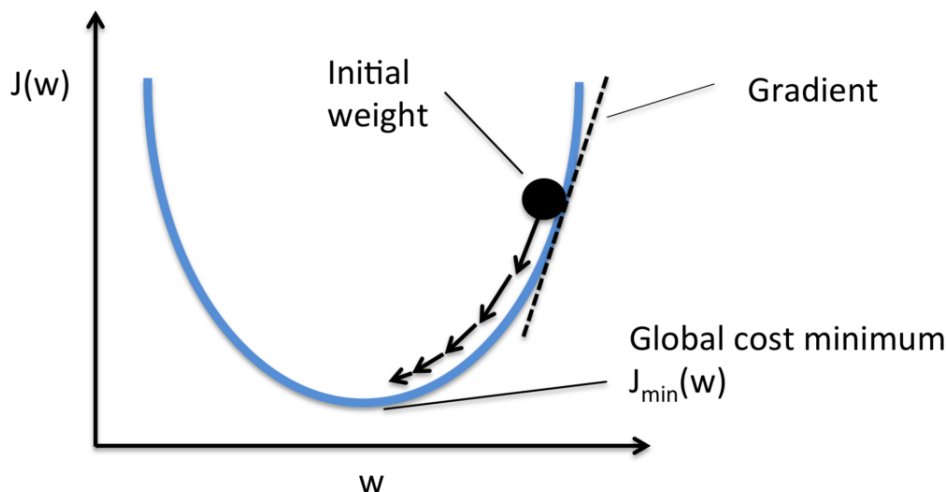


Figura 2.6: Representação da Descida do Gradiente para uma variável.

Na figura 2.6, tem-se um peso inicial (*Initial weight*), cujo valor otimizado se encontra em $J_{\min}(w)$. Para chegar até esse valor mínimo, calcula-se o gradiente em seu ponto inicial e o gradiente indicará que o peso deve se mover para a esquerda, por um fator igual à taxa de aprendizado. Após a atualização, o peso poderá estar em seu valor mínimo, o que encerraria o algoritmo, poderá estar ainda a direita do seu objetivo, o que fará com o que o gradiente indique ele ainda deve se mover para a esquerda, porém como ele estará mais perto, o gradiente terá um menor valor, indicando uma movimentação mais suave, ou poderá estar a esquerda do valor mínimo, nesse caso o gradiente indicará que ele deve agora se mover no sentido contrário.

O conceito de taxa de aprendizado é importante ser mencionado, porque é através dessa taxa que é ajustada a velocidade com que os pesos serão ajustados, taxas de aprendizagem muito grande, farão com que os pesos sejam atualizados muito rápido, fazendo com que eles fiquem ultrapassando seu valor mínimo, seja para a direita ou para a esquerda, fazendo, ainda, uma analogia com a figura 2.6. Taxas de aprendizado muito pequenas farão com que os pesos sejam ajustados muito lentamente, de forma que o treinamento das redes poderá levar muito tempo.

Sendo assim, esse é um parâmetro que deve ser escolhido com muita cautela.

Pode-se modelar a descida do gradiente de forma simplificada da seguinte forma,

Dada uma função de custo dos coeficientes,

$$custo = f(w_i, b_i) \quad (2.11)$$

Calcula-se a derivada da função de custo,

$$custo' = derivada(f(w_i, b_i)) \quad (2.12)$$

Dada a derivada da função de custo e uma taxa de aprendizado α ,

$$(w_{i+1}, b_{i+1}) = (w_i, b_i) - (\alpha \cdot custo') \quad (2.13)$$

O processo apresentado pelas equações 2.11, 2.12 e 2.13 é então repetido, até que a função de custo dos coeficientes atinja seu valor mínimo ou seja tão próxima de seu valor mínimo que satisfaça o problema.

O algoritmo de descida do gradiente é simples, porém extremamente poderoso, quando é generalizado para números muito grandes de coeficientes, como usualmente ocorre em redes neurais profundas.

BackPropagation

O *backpropagation* tem por objetivo propagar o erro, também chamado de função de custo, por todos os pesos da rede, partindo da saída indo em direção a entrada. Essa propagação é feita através do cálculo do gradiente do erro, ou custo, porém o cálculo desse gradiente de forma numérica pode ser extremamente custoso computacionalmente falando. O surgimento do algoritmo do *backpropagation* se deu para contornar essa barreira.

Para entender o *backpropagation* é necessário conhecer a regra da cadeia do cálculo. Ela é utilizada para calcular as derivadas de funções formadas por composições de outras funções, cujas derivadas são conhecidas. O *backpropagation* nada mais é do que um algoritmo que calcula essas derivadas de forma muito eficiente.

Dado um x real, tal qual f e g são ambas funções mapeando de um número real a outro número real. Fazendo-se $y = g(x)$ e $z = f(g(x)) = f(y)$. Tem-se que a regra da cadeia é dada por:

$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx} \quad (2.14)$$

Generalizando essa equação além do caso escalar, pode-se imaginar $x \in R_m$, $y \in R_n$, g mapeando de R_m para R_n e f mapeando de R_n para R . Então se $y = g(x)$ e $z = f(y)$,

$$\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_j} \cdot \frac{\partial y_j}{\partial x_i} \quad (2.15)$$

Com uma notação vetorial, escreve-se:

$$\nabla x^z = \left(\frac{\partial y}{\partial x} \right)^T \cdot \nabla y^z \quad (2.16)$$

onde $\partial y / \partial x$ é a matriz Jacobiana de g .

A partir da relação na equação 2.16, nota-se que o gradiente de uma variável x , pode ser obtido pela multiplicação da matriz Jacobiana $\partial y / \partial x$ pelo gradiente ∇y^z . O algoritmo de *backpropagation* consiste em realizar essas operações de

multiplicação do gradiente pelo jacobiano para cada nó da rede.

Redes Neurais Recorrentes

Uma característica importante a ser mencionada das redes não recorrentes são o fato de aprenderem cada passo de informação de forma individual e é nesse ponto que as redes recorrentes diferem, uma vez que em suas arquiteturas, algumas camadas possuem *loops* de *feedback*, onde informações de um passo são realimentadas ao passo anterior, de forma a criar um conceito de memória.

Essa característica de *loop* das redes neurais recorrentes (DE CARVALHO *et al.*, 2018, LUKOŠEVIČIUS e JAEGER, 2009) faz com que elas sejam muito boas para o aprendizado de dados sequenciais, como textos, sons, dados de séries temporais e qualquer outro tipo de informação que possa ser modelada como uma sequência.

O modelo de redes recorrentes possui um conceito chamado de estado oculto, que é a informação do estado em que a rede está. O estado oculto é passado para a própria camada no *loop* de *feedback* e combinado com os dados recebidos da camada anterior, formando um novo estado oculto e uma nova classificação. Em resumo, o papel do estado oculto é ser combinado com os dados de entrada da camada em questão para criar um novo estado oculto, abandonando o anterior e gerando uma nova classificação, mais precisa do que a anterior.

Pode-se fazer uma analogia do estado oculto com o estado de consciência da memória humana, uma vez que no pensamento humano, uma informação será interpretada de diferentes formas dependendo do contexto em que se apresenta, na prática o estado oculto das redes recorrentes tem a mesma função.

As redes neurais recorrentes se mostraram extremamente promissoras no aprendizado de informações sequenciais, porém devido a sua característica recorrente, na qual um estado oculto é armazenado com informações de estados anteriores, o problema do desaparecimento do gradiente as torna extremamente difíceis de serem treinadas, uma vez que a cada nova iteração, a informação armazenada do estado anterior é em parte perdida, com isso, cerca de 4 a 8 novas iterações são o suficiente para que a informação de um estado seja insignificante, fazendo com que a principal característica da recorrência seja perdida.

Problema do Desaparecimento do Gradiente O problema matemático que mais dificulta o treinamento das redes neurais profundas, em especial as recorrentes, é o problema do desaparecimento do gradiente (HU *et al.*, 2019, SQUARTINI *et al.*, 2003).

Para entender o porquê do acontecimento do desaparecimento do gradiente, é necessário entender como funciona a propagação do gradiente pelas camadas da rede neural.

De forma a simplificar o entendimento, imagina-se uma rede neural profunda, com três camadas internas e apenas um neurônio em cada camada,

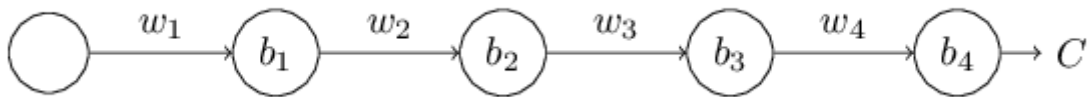


Figura 2.7: Rede Neural Profunda Simplificada.

onde,

- w_i são os pesos
- b_i são os vieses (*bias*)
- C é uma função de custo qualquer

O problema do desaparecimento do gradiente é relevante nas primeiras camadas das redes, então a exemplificação se dará no gradiente da primeira camada oculta, ou seja, $\partial C / \partial b_1$.

As notações utilizadas nos cálculos serão,

- $\partial C / \partial b_1$: Derivada parcial de C em relação a b_1
- a_i : Valor de saída no neurônio i
- z_i : Valor de entrada no neurônio i , $z_i = w_i \cdot a_i + b_i$
- σ : Função de ativação
- σ' : Derivada da função de ativação

Dadas as notações, pode-se definir a expressão para $\partial C/\partial b_1$.

$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \times w_2 \times \sigma'(z_2) \times w_3 \times \sigma'(z_3) \times w_4 \times \frac{\partial C}{\partial a_4} \quad (2.17)$$

É importante observar que para cada neurônio na rede, existe um termo $\sigma'(z_i)$ e um peso w_i , pois isso exemplifica a escalabilidade entre esses produtos a cada vez que a rede possui mais camadas.

Fazendo uma pequena mudança Δb_1 no viés b_1 , um efeito em cascata ocorrerá, causando primeiro uma pequena mudança na saída do primeiro neurônio oculto a_1 e assim por diante até chegar à função de custo, resultando em,

$$\frac{\partial C}{\partial b_1} \approx \frac{\Delta C}{\Delta b_1} \quad (2.18)$$

A partir da relação da equação 2.18 é possível escrever uma relação para $\partial C/\partial b_1$ a partir de cada etapa da cascata. Para chegar a essa expressão, dá-se que uma pequena variação Δb_1 no viés b_1 implica numa variação Δa_1 na saída do primeiro neurônio oculto de,

$$\begin{aligned} \Delta a_1 &\approx \frac{\partial \sigma(w_1 \cdot a_0 + b_1)}{\partial b_1} \cdot \Delta b_1 \\ &= \sigma'(z_1) \cdot \Delta b_1 \end{aligned} \quad (2.19)$$

A partir da equação 2.19, fica comprovado que uma pequena variação Δb_1 implica numa pequena variação Δa_1 na saída do primeiro neurônio oculto. Propagando essa variação em a_1 , tem-se que,

$$\begin{aligned} \Delta z_2 &\approx \frac{\partial z_2}{\partial a_1} \cdot \Delta a_1 \\ &= w_2 \cdot \Delta a_1 \end{aligned} \quad (2.20)$$

Com as equações 2.19 e 2.20 pode-se isolar os termos $\sigma'(z_1) \cdot \Delta b_1$ e w_2 que são o primeiro e segundo termos da equação 2.18, respectivamente. Combinando as equações 2.19 e 2.20, chega-se a relação que mostra a influência da variação no viés b_1 no termo z_2 ,

$$\Delta z_2 \approx \sigma'(z_1) \cdot w_2 \cdot \Delta b_1 \quad (2.21)$$

A partir da equação 2.21 é possível notar que replicando os algebrismos apresentados nas equações 2.19 e 2.20 para os demais termos a_i e w_i , chega-se a seguinte expressão,

$$\Delta C \approx \sigma'(z_1) \cdot w_2 \cdot \sigma'(z_2) \cdot w_3 \cdot \sigma'(z_3) \cdot w_4 \cdot \sigma'(z_4) \cdot \frac{\partial C}{\partial a_4} \cdot \Delta b_1 \quad (2.22)$$

Dividindo toda a equação 2.22 por Δb_1 ,

$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \cdot w_2 \cdot \sigma'(z_2) \cdot w_3 \cdot \sigma'(z_3) \cdot w_4 \cdot \sigma'(z_4) \cdot \frac{\partial C}{\partial a_4} \quad (2.23)$$

Com a equação 2.23 fica demonstrado matematicamente a influência que uma pequena variação no viés b_1 tem em toda a cadeia de propagação. Observando a equação 2.23 tem-se que com exceção do último termo, toda a expressão é formada por produtos entre $w_i \cdot \sigma'(z_i)$. A notação σ representando a função de ativação é usualmente uma sigmoide, onde sua derivada é representada no gráfico da figura 2.8.

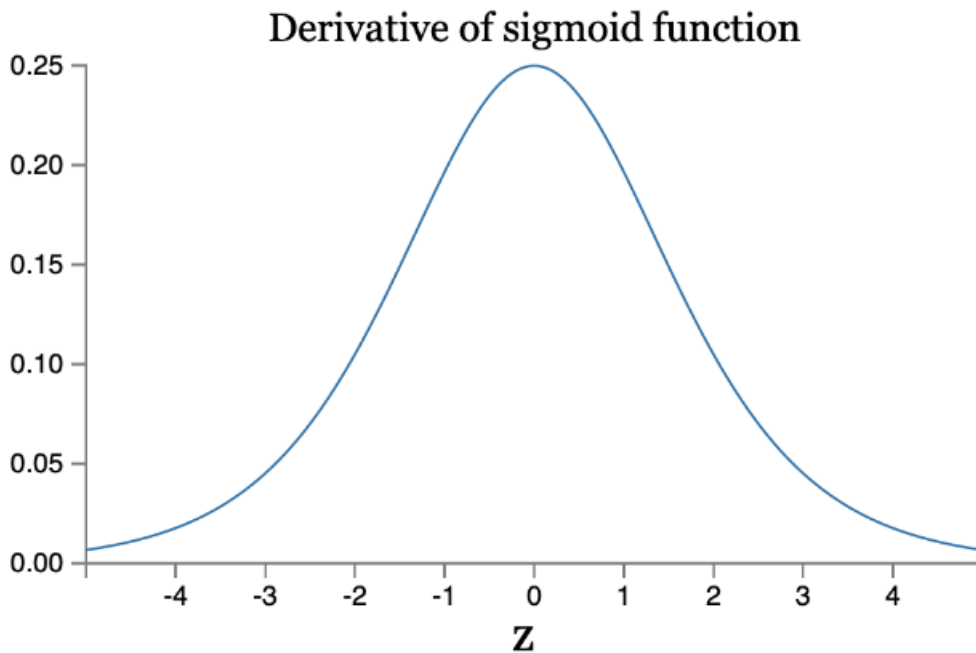


Figura 2.8: Derivada da Função Sigmoide.

Na literatura é indicado inicializar os pesos das redes neurais com uma distribuição normal com média zero e desvio padrão igual a 1, ou seja, todos os w_i terão valores absolutos menores do que 1. Como é visto a partir do gráfico, a derivada da função sigmoide $\sigma'(z)$ atinge seu valor máximo igual a 0.25 para valores de $z = 0$. Dessa forma, a equação 2.23 passa a ter uma cascata de multiplicações de valores

menores do que 1, o que faz com que seu valor final tenha uma tendência muito forte de convergência para zero a cada vez que uma nova camada oculta é inserida.

Exemplificando de forma explícita a mesma equação mostrada em 2.23, porém para $\partial C/\partial b_3$,

$$\frac{\partial C}{\partial b_3} = \sigma'(z_3) \cdot w_4 \cdot \sigma'(z_4) \cdot \frac{\partial C}{\partial a_4} \quad (2.24)$$

é possível ver a relação existente entre as equações, porém a relação apresentada para $\partial C/\partial b_1$, ou seja, para o primeiro neurônio, possui dois termos multiplicativos extras, indicando que uma variação na função de custo terá uma influência menor no ajuste dos pesos nas primeiras camadas.

Essa formulação matemática não é prova de que o desaparecimento do gradiente obrigatoriamente ocorrerá, mas ela mostra o porque ele acontece e exemplifica que conforme o número de camadas ocultas cresce, maiores probabilidades do problema ocorrer.

Diferentes formas de modelar redes neurais recorrentes surgiram e foram testadas, umas delas foi a *Long-Short Term Memory* (LSTM), que surgiu como uma forma de solucionar o problema do desaparecimento do gradiente e que é considerada nos dias atuais o estado da arte em redes neurais recorrentes.

Long-Short Term Memory As LSTM foram propostas nos anos 90, pelos pesquisadores alemães Sepp Hochreiter e Juergen Schmidhuber (HOCHREITER e SCHMIDHUBER, 1997), como uma solução ao problema do desaparecimento do gradiente e da perda da memória de estado de recorrência. Essas redes tem como característica preservar o erro de uma etapa de *feedforward* copiando-o por diversas camadas e épocas de treinamento. Mantendo um erro mais constante é possível que a rede continue aprendendo durante muitos passos de tempo, comprovadamente, mais de 1000 passos.

O principal conceito das redes LSTM, são as células de memória, que fazem o papel do corpo do neurônio perceptron. Essas células possuem um conceito de portões, onde são controladas as permissões ou negações de entrada, saída e esquecimento de informações. Diferente do armazenamento digital de informação em computadores, esses portões são analógicos, implementados com funções sigmóides (HAN e MORAGA), 1995, que levam todos os sinais para uma escala entre 0 e 1.

Além da vantagem de criar um sinal analógico nos portões, ainda tem-se a característica da função sigmóide ser diferenciável, o que a torna adequada para o algoritmo de *backpropagation*.

Na figura 2.9 é dada uma ilustração de uma célula LSTM e seus componentes,

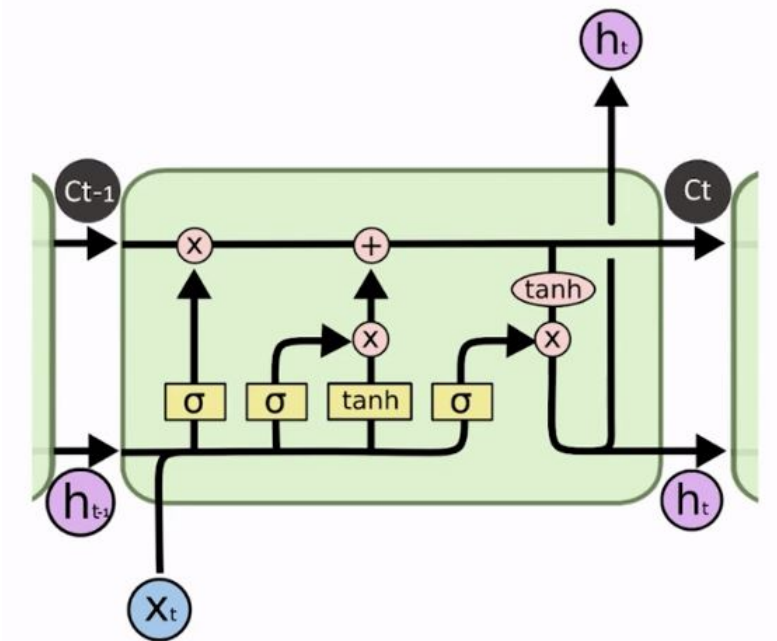


Figura 2.9: Célula LSTM (OLAH, 2015).

Para um melhor entendimento, tem-se as notações utilizadas na figura e que serão utilizadas na construção da formulação matemática,

- X_t : Entrada de informação na célula;
- h_t : Um dos estados da célula, que é passado para ela mesma e também funciona como saída, para que a informação seja passada à próxima célula da rede;
- C_t : O segundo estado da célula, apenas interno;
- C_{t-1} e h_{t-1} : Estados da célula na passagem anterior;
- b_f : viés (*bias*);
- W_f : Peso;
- σ : Função de Ativação Sigmoides;
- \tanh : Função de Ativação Tangente Hiperbólica;

- \times : Operação de multiplicação;
- $+$: Operação de soma.

Dadas as notações, pode-se partir para o entendimento da funcionalidade da célula LSTM. Existem duas conexões horizontais, representando os estados das células, a superior, partindo de C_{t-1} para C_t representando o estado interno e a inferior partindo de h_{t-1} para h_t representando o estado que é passado para a próxima célula, ou seja, saída de informação da célula em questão. É importante entender que a memória de uma rede neural LSTM é representada pelas informações armazenadas nesses estados de célula, uma vez que é esse estado que é mantido e atualizado junto com o novo *input* recebido.

Em uma LSTM existem os conceitos de portões, que funcionam como filtros, e são representados na figura 2.9 pelas linhas verticais saindo das funções de ativação. A função desses portões é filtrar o sinal recebido tanto da célula anterior X_t junto com o sinal recebido recursivamente pela própria célula h_t , passando adiante apenas a porcentagem relevante desse sinal.

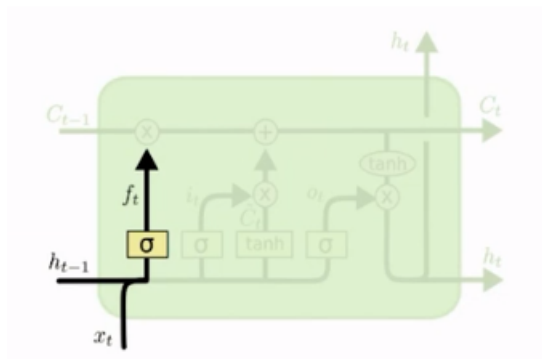


Figura 2.10: Portão de Esquecimento, Célula LSTM (OLAH, 2015).

Na figura 2.10 fica representado o primeiro portão da célula, o portão de esquecimento. Esse portão é alimentado com as entradas X_t e h_{t-1} e essas informações são multiplicadas por um peso e somadas a um viés, em seguida filtradas pela função sigmoide, gerando uma saída f_t , que por ser uma saída de sigmoide, terá um valor entre 0 e 1, representando a porcentagem de informação desse sinal que deverá ser esquecida.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.25)$$

O f_t será utilizado na atualização do estado C_t , mais adiante.

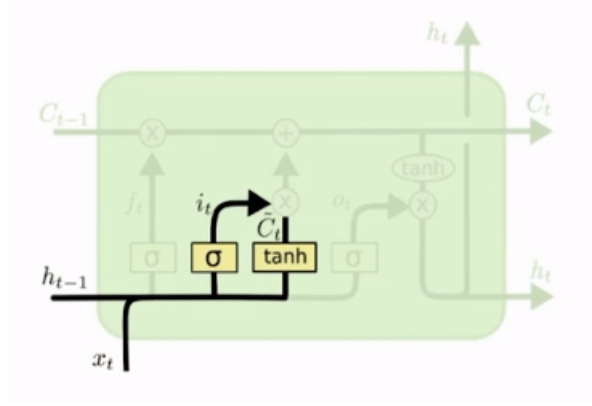


Figura 2.11: Portão de Entrada, Célula LSTM (OLAH, 2015).

Na figura 2.11, o segundo portão, portão de entrada, é representado, onde são geradas as novas informações a serem armazenadas na célula, etapa \tanh e o quanto dessa nova informação deve ser armazenada, etapa σ .

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.26)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.27)$$

Com o novo estado \tilde{C} , criado na equação 2.27, o estado anterior C_{t-1} e o sinal de esquecimento f_t , atualiza-se a informação do estado C ,

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (2.28)$$

Na equação 2.28, tem-se a expressão final de C_t , que é a expressão para a memória de longo prazo da LSTM. A característica de a memória de longo prazo vinda de uma célula anterior C_{t-1} sempre ser multiplicada por uma f_t faz com que esse estado de memória possa ser preservado indefinidamente, dependendo apenas da calibração do portão de esquecimento.

A memória de curto prazo, por sua vez, é representada pelo estado h , que é o fator passado entre células e que tem por característica participar de todas as operações matemáticas na célula, tendo influência inclusive na criação do novo estado \tilde{C}_t que será acrescido à memória de longo prazo.

Para completar o entendimento da célula LSTM e formular a expressão para o cálculo de h_t , é necessário passar pelo último portão, o portão de saída.

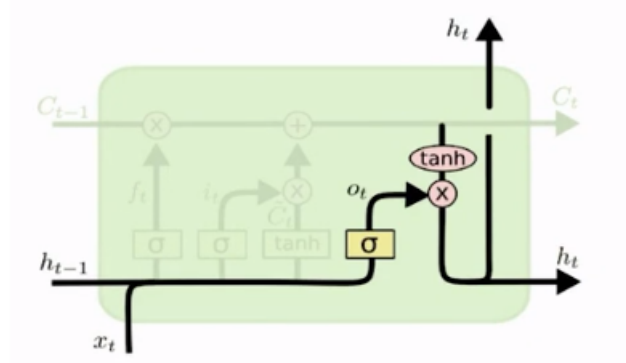


Figura 2.12: Portão de Saída, Célula LSTM (OLAH, 2015).

Na figura 2.12 tem-se uma pequena inversão do fluxo de informação, onde pela primeira vez uma informação do trajeto de C flui para o trajeto de h . Sendo assim, h_t nada mais é do que o produto da memória de longo prazo C_t atenuada por uma função de ativação tangente hiperbólica com os sinais x_t e h_{t-1} filtrados pela sigmoide. Dessa forma, a memória de curto prazo h_t é uma combinação da memória de longo prazo, com os sinais de operação de curto prazo.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \quad (2.29)$$

Dada essa expressão, finalmente combinando as equações 2.28 e 2.29, chega-se a expressão para h_t , que tem por função ser passada novamente a essa célula, como memória de curto prazo e à próxima célula, como novo *input*

$$h_t = o_t.tanh(C_t) \quad (2.30)$$

Entendendo o funcionamento da célula LSTM, é importante retornar a questão do desaparecimento do gradiente e sua formulação, apresentada na seção 2.5.5. Ficou entendido que aplicações sucessivas do gradiente em redes com muitas camadas, o gradiente tende a desaparecer para as primeiras camadas da rede. Para estudar esse efeito na LSTM, resgata-se a equação 2.28 para a memória de longo prazo.

Reescrevendo a equação 2.28, substituindo suas partes das equações 2.25, 2.26 e 2.27,

$$C_t = \sigma(W_f.[h_t, x_t] + b_f).C_{t-1} + \sigma(W_i.[h_t, x_t] + b_i).tanh(W_c.[h_t, x_t] + b_c) \quad (2.31)$$

Aplicando o gradiente $\partial s_i / \partial s_{i-1}$ na equação acima,

$$\frac{\partial C_t}{\partial C_{t-1}} = \sigma(W_f \cdot [h_t, x_t] + b_f) \cdot C_{t-1} + \frac{\partial}{\partial C_{t-1}} \sigma(W_i \cdot [h_t, x_t] + b_i) \cdot \tanh(W_c \cdot [h_t, x_t] + b_c) \quad (2.32)$$

ou seja,

$$\frac{\partial C_t}{\partial C_{t-1}} \approx \sigma(W_f \cdot [h_t, x_t] + b_f) \cdot C_{t-1} \quad (2.33)$$

Dessa forma, pode-se notar que o gradiente comporta-se praticamente de forma similar ao portão de esquecimento, sendo esse o mais importante componente na solução do problema do desaparecimento do gradiente nas redes neurais LSTM (PINHEIRO, 2020).

Capítulo 3

O Problema e a Metodologia Proposta

Dentro do contexto da extensão de vida útil da usina e diante da necessidade de um estudo de envelhecimento dos equipamentos elétricos e eletrônicos de seu prédio de contenção, torna-se necessário o conhecimento das temperaturas de operação da usina em todos os seus anos de operação. Dentro deste contexto, esta pesquisa visa propor um método de simulação de sensores de temperatura virtuais, tendo por objetivos individuais fazer a inferência da temperatura em uma dada posição em um dado tempo no passado.

Os dados disponíveis para a elaboração desse trabalho são dados de sensores instalados em posições estratégicas no interior do prédio de contenção da usina a partir do ano de 2015. Esses sensores representam as posições onde deseja-se conhecer as temperaturas em todos os anos de operação e serão aqui apresentados como Sensor-X, onde X representa um número de 1 a 82. Cada um desses sensores possui uma coordenada tridimensional (x, y, z) representando a exata posição onde ele foi instalado no interior do prédio de contenção.

Além dos dados dos Sensores-X, que no escopo do aprendizado das redes neurais serão os dados alvo dos treinamentos, tem-se conhecimento dos dados de temperatura dos detectores do Sistema Integrado de Computadores de Angra (SICA), os quais possuem posições estratégicas para a monitoração dos estados de operação, mas não são estrategicamente favoráveis ao escopo da extensão de vida útil.

Sendo assim, deseja-se modelar redes neurais que, recebendo as informações dos dados de temperatura dos medidores do SICA, consigam mapear as temperaturas dos sensores-X, de forma que possa ser feita uma simulação de medidores virtuais

que façam a inferência das temperaturas para a obtenção do envelhecimento dos equipamentos elétricos, informação essencial à elaboração do programa de qualificação ambiental.

3.1 Seleção de Variáveis

Apesar de as redes neurais serem capazes de um trabalho muito eficiente na previsão de informações, elas são totalmente dependentes de dados, mas não basta que sejam quaisquer dados. As redes neurais, assim como todo algoritmo de *machine learning* são muito eficientes em encontrar relações entre os dados de entrada e saída e com isso são capazes de realizar previsões e inferências.

Partindo do princípio de que os algoritmos de aprendizado em sua essência apenas encontram a relação entre os dados de entrada e saída, os dados fornecidos como entrada para as redes neurais precisam ter alguma relação com os dados fornecidos como saída para as redes e quanto melhor for essa relação entre dados de entrada e saída, maior a eficiência das redes neurais em aprender e mais precisas serão suas previsões ou inferências.

Diante disso a etapa de seleção de variáveis em qualquer trabalho relacionado a aprendizado de máquina, é de suma importância para otimizar a eficiência dos algoritmos.

Usualmente, em problemas de análise de dados, métodos de seleção de variáveis conhecidos na literatura, como a floresta aleatória (CHEN e ISHWARAN, 2012) são utilizados. Esses métodos calculam a correlação entre cada variável e selecionam as que possuem melhores correlações entre si, o que aumenta a eficiência das redes neurais, uma vez que quanto maior a relação entre as variáveis de entrada e de saída, mais uma rede neural é capaz de aprender.

Como o problema de seleção de variáveis deste trabalho é relativamente simples, uma vez que a única variável a se considerar é a temperatura, foi proposta uma seleção de variáveis utilizando uma distância euclidiana mínima entre os sensores que seriam utilizados em treinamento (sensores do SICA) e os sensores alvo (sensores-X). Através de um mapeamento da planta baixa da usina, foram retiradas as coordenadas x, y, z de cada um dos sensores utilizados neste trabalho e através de um cálculo simples de distância,

$$d = \sqrt{x^2 + y^2 + z^2} \quad (3.1)$$

Foram feitos pares entre treinamento e alvo, a partir dos respectivos equipamentos com menor d .

3.2 Pré-Processamento dos dados

Os ciclos de operação das usinas nucleares são contemplados com intervalos regulares para troca de combustível, sendo assim um ciclo completo de operação é considerado desde o momento do religamento pós troca de combustível até o desligamento para a troca subsequente. Entendendo o conceito de ciclo de operação, pode-se entender que no escopo deste trabalho, foram selecionados dois ciclos para análise, contemplando os anos de 2017-2018 e 2018-2019.

É comum a todos os dados reais, que existam ruídos e/ou *outliers* (valores extremos) e não é diferente no caso das temperaturas da usina. Sendo assim, um tratamento de *outliers* foi necessário para evitar picos repentinos na série temporal de temperaturas.

Pensando no conceito de ciclos de operação, em seu início e seu fim existe um comportamento claro das temperaturas no interior do prédio de contenção. Quando a usina está desligada para a realização da troca de combustível, há baixa produção de calor pelo reator, sendo assim a temperatura é mais baixa em todo o prédio. Ao religar a usina, a temperatura se eleva junto com o aumento de potência do reator, até que este atinja máxima potência, fazendo com que a temperatura atinja estabilidade. O mesmo ocorre no momento do desligamento, com a temperatura caindo, ao passo que a potência do reator vai sendo diminuída até desligar.

Uma vez que o processo de aumento e de diminuição de potência do reator não ocorre repentinamente, para uma análise mais uniforme dos *outliers* nos dados de temperatura foram removidos dos dados os registros de 3 semanas após o religamento e 3 semanas antes do desligamento da usina em cada ciclo, de forma que os dados resultantes fossem apenas referentes ao período de operação à plena potência.

Após essa primeira preparação, foi feito um tratamento de *outliers* utilizando uma análise estatística com a distribuição Gaussiana. Uma vez que os dados sendo trabalhados são dados reais e possuem dezenas de milhares de registros, pode-se assumir que a distribuição de temperatura segue uma distribuição Gaussiana. Sendo assim, foram calculados os parâmetros referentes a distribuição normal para cada um dos sensores, tanto do SICA, quanto dos Sensores-X.

Em uma distribuição Gaussiana perfeita, 99,7% dos valores estão dentro do intervalo da média ± 3 desvios padrão e 95% dentro do intervalo da média ± 2 desvios padrão. Utilizando esse conceito estatístico, foram feitos dois filtros. No primeiro, todos os valores fora da faixa de 2 desvios padrão foram removidos e após essa etapa, calculou-se novamente a média e desvio padrão do *dataset* resultante e foram passados pelo segundo filtro, onde de forma mais seletiva, apenas os dados fora do intervalo de 3 desvios padrão foram removidos. Fazendo uma análise visual dos gráficos (figuras 3.1 e 3.2) dos dados antes e depois dos filtros, é possível ver o impacto do tratamento de *outliers*.

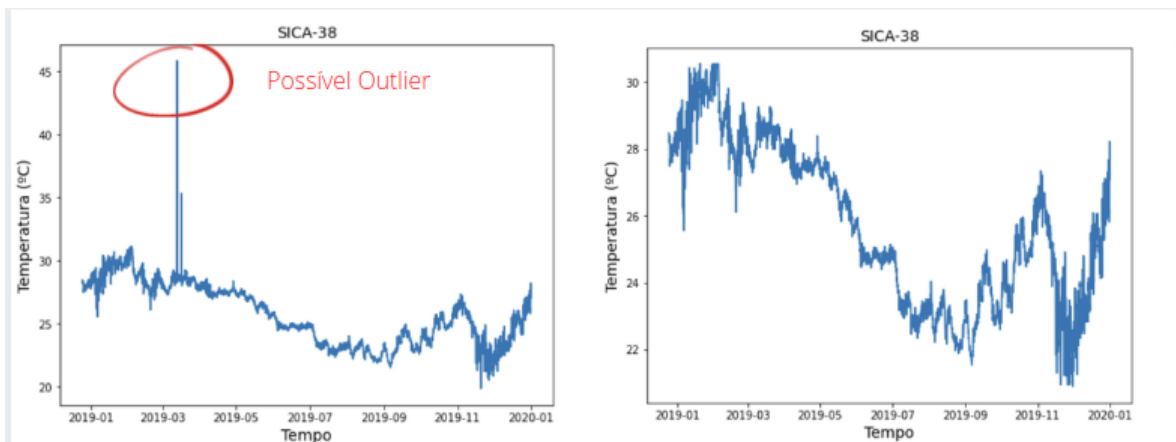


Figura 3.1: Sensor SICA-38. À esquerda, gráfico antes do tratamento de *outlier*, à direita, após o tratamento.

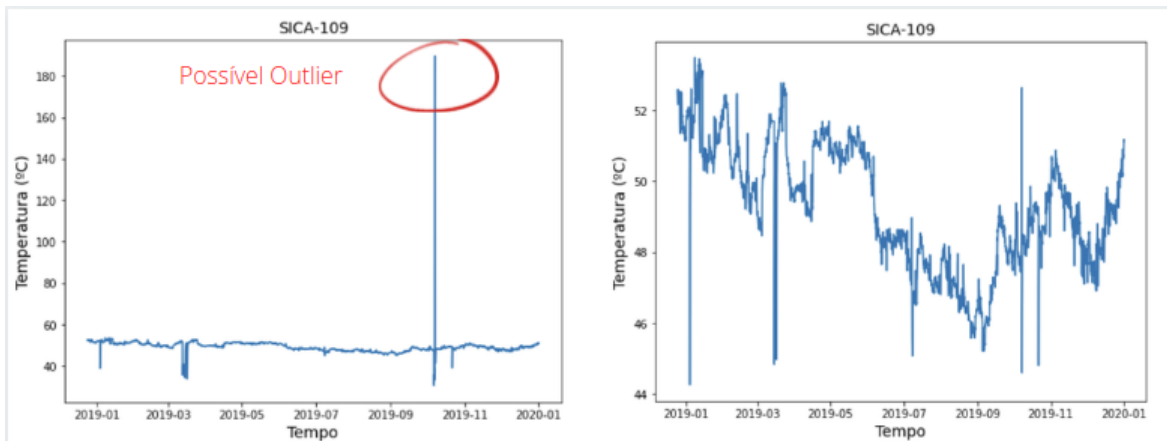


Figura 3.2: Sensor SICA-109. À esquerda, gráfico antes do tratamento de *outlier*, à direita, após o tratamento.

3.3 Modelagem da Rede Neural

A modelagem proposta, parte do princípio de que as redes neurais recebem como entrada as temperaturas dos sensores SICA-X e as saídas esperadas são os valores de temperatura dos Sensores-X, sendo assim um erro é calculado como dimensão da diferença entre a saída encontrada e a saída esperada e esse erro utilizado na correção do valor dos pesos. A figura 3.3 exemplifica essa modelagem.

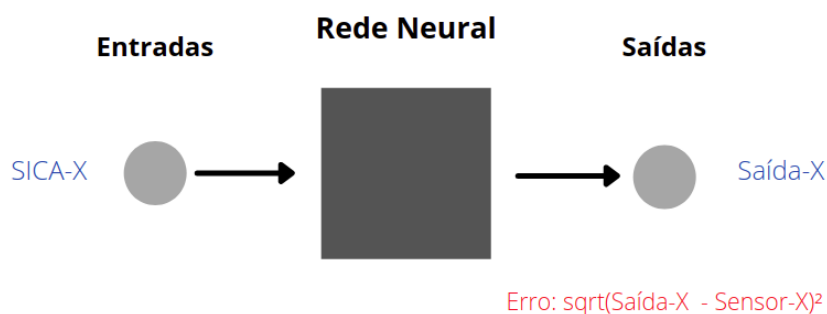


Figura 3.3: Representação das entradas e saídas da rede neural na metodologia proposta.

A modelagem de redes neurais, usualmente passa por um processo empírico, uma vez que vários modelos são testados a fim de encontrar aquele que melhor entrega solução para o problema em questão. Cada modelagem passa desde a escolha da arquitetura da rede até a escolha dos hiperparâmetros que serão utilizados. Neste trabalho, foram utilizadas duas arquiteturas, mencionadas anteriormente,

- DNN;
- LSTM.

E como hiperparâmetros, foram ajustados,

- Número de épocas;
- Número de camadas internas,
- Número de neurônios por camada interna;

Além dos hiperparâmetros que foram alterados no trabalho, inúmeros outros podem ser configurados, como por exemplo a função de ativação, inicializador de viés, inicializador de kernel, porém foram mantidos constantes, por serem parâmetros onde a literatura indica terem valores mais eficientes.

Para o escopo do projeto, os vieses foram inicializados com zero, o inicializador de kernel utilizado foi *glorot uniform* (AN YOSHUA BENGIO, 2010) e a função de ativação, para a DNN foi utilizada Relu e para a LSTM Sigmoide e Tangente Hiperbólica, como é padrão na célula LSTM. Como função de custo, foi escolhida a função de erro médio quadrático.

Partindo do principio de que vários modelos precisam ser criados para fazer uma seleção de qual arquitetura e quais hiperparâmetros melhor resolvem o problema, foi escolhido começar pelas modelagens mais simples, sendo assim, as DNN.

3.3.1 Modelagem da DNN

Para a DNN, como dito no item 3.3, foram treinados vários modelos com diferentes hiperparâmetros. Seu treinamento, assim como o da LSTM, que será visto mais a frente, foi feito nos dados do ciclo de operação 2018-2019.

Os dados, preparados como descrito no item 3.2, foram então separados em dados de treino (64% do total dos dados), validação (16%) e teste (20%). Os dados de teste não são apresentados a rede durante o treinamento, servindo como dados para fazer uma avaliação ao final do treinamento. Os dados de validação são passados a rede durante o treinamento, de forma a rede conseguir melhorar o treinamento, de acordo com a performance dos dados de validação.

Cada modelo é representado por um conjunto de parâmetros e dentro deste mesmo modelo são treinadas várias redes neurais, uma para cada Sensor-X. Após o treinamento de uma rede para cada Sensor-X é feita uma avaliação do erro individual de cada uma e então calculado o erro médio para aquele modelo.

Após vários treinamentos com diferentes modelos de parâmetros, aquele que obteve o melhor resultado, observando o erro médio de cada rede individual, foi escolhido para uma análise pós-treinamento. Os parâmetros utilizados neste modelo são apresentados na tabela 3.1.

Tabela 3.1: Parâmetros Utilizados no melhor modelo de DNN.

Parâmetro	Valor
Número de Épocas	200
Número de Camadas Internas	7
Número de Neurônios por Camada	[64, 128, 256, 256, 256, 128, 64]

Resultados da DNN

Nesta etapa, foram treinadas 36 redes utilizando cada um dos modelos. Para o modelo, cujos parâmetros foram apresentados na tabela 3.1, as avaliações de cada rede são apresentadas na tabela 3.2,

Tabela 3.2: Erros do Modelo DNN.

Sensor	Erro em Treinamento	Erro em Validação	Erro em Teste
Sensor-16	0.4683	0.4535	0.4610
Sensor-18	0.3968	0.3815	0.3968
Sensor-20	0.7070	0.7422	0.6963
Sensor-22	0.2409	0.2327	0.2337
Sensor-23	0.1527	0.1415	0.1540
Sensor-24	0.1479	0.1457	0.1443
Sensor-28	0.3183	0.3091	0.3312
Sensor-30	0.4812	0.4998	0.5051
Sensor-33	5.9251	6.0015	6.0314
Sensor-34	6.6827	6.7994	6.7325
Sensor-35	8.2844	8.3852	8.3753
Sensor-36	1.7580	1.8686	1.9216

Continua na próxima página

Tabela 3.2 – Erros do Modelo DNN.

Sensor	Erro em Treinamento	Erro em Validação	Erro em Teste
Sensor-37	0.8808	0.8141	0.8438
Sensor-38	1.0066	0.9593	1.0035
Sensor-40	1.2286	1.2521	1.2496
Sensor-41	4.5336	4.5069	4.7922
Sensor-45	10.2935	10.9506	9.9176
Sensor-48	6.1892	6.1745	6.0771
Sensor-49	5.7023	6.1230	5.7141
Sensor-51	9.6361	10.2444	9.4872
Sensor-52	0.8637	0.9266	0.8566
Sensor-54	0.6070	0.6030	0.5818
Sensor-55	6.8022	6.7767	6.8720
Sensor-57	10.2188	9.5807	9.6646
Sensor-58	0.5580	0.5642	0.5644
Sensor-63	1.8003	1.7923	1.7855
Sensor-66	0.9083	0.9450	0.9353
Sensor-67	0.7919	0.8024	0.7914
Sensor-72	0.6429	0.6218	0.6212
Sensor-74	0.1801	0.1758	0.1732
Sensor-75	0.1915	0.1762	0.1871
Sensor-77	5.1582	5.0013	5.3110
Sensor-78	0.1779	0.1854	0.1762
Sensor-79	0.7197	0.7161	0.6951
Sensor-80	5.1885	5.2934	5.2031
Sensor-82	3.8543	4.0034	3.8357

As estatísticas referentes a esse modelo, são apresentadas na tabela 3.3

Tabela 3.3: Estatísticas do Modelo DNN.

	Erro em Treinamento	Erro em Validação	Erro em Teste
Contagem	36	36	36
Média	2.8804	2.9209	2.8701
Desvio Padrão	3.2682	3.3306	3.2168
Valor Mínimo	0.1478	0.1450	0.1443
Valor Máximo	10.2935	10.9506	9.9176

Os parâmetros utilizados na comparação entre diferentes modelos, foram as médias em treino, validação e teste, como apresentados para o melhor modelo DNN na tabela 3.3. Vale observar que entre os valores mínimo e máximo de uma rede individual, existe uma grande diferença de erro, o que faz o desvio padrão ficar elevado.

Após a avaliação dos parâmetros e dos erros do melhor modelo escolhido para a DNN, foram feitas as previsões nos dados do ciclo de operação 2017-2018, que são dados que as redes nunca viram antes, assim como os dados de teste, porém com o diferencial de que os dados de um ciclo de operação diferente possuem diferentes correlações entre Sensor-X e SICA-X.

Entendendo como objetivo do trabalho, que as redes neurais treinadas em um ciclo de operação sejam capazes de generalizar a previsão de temperaturas para todos os ciclos de operação passados, é importante que a previsão feita no ciclo de operação 2017-2018 dê resultados satisfatórios.

Algumas das previsões feitas no ciclo de 2017-2018, são apresentadas a seguir, junto com a avaliação pelo erro quadrático médio em relação ao valor real, medido pelo Sensor-X.

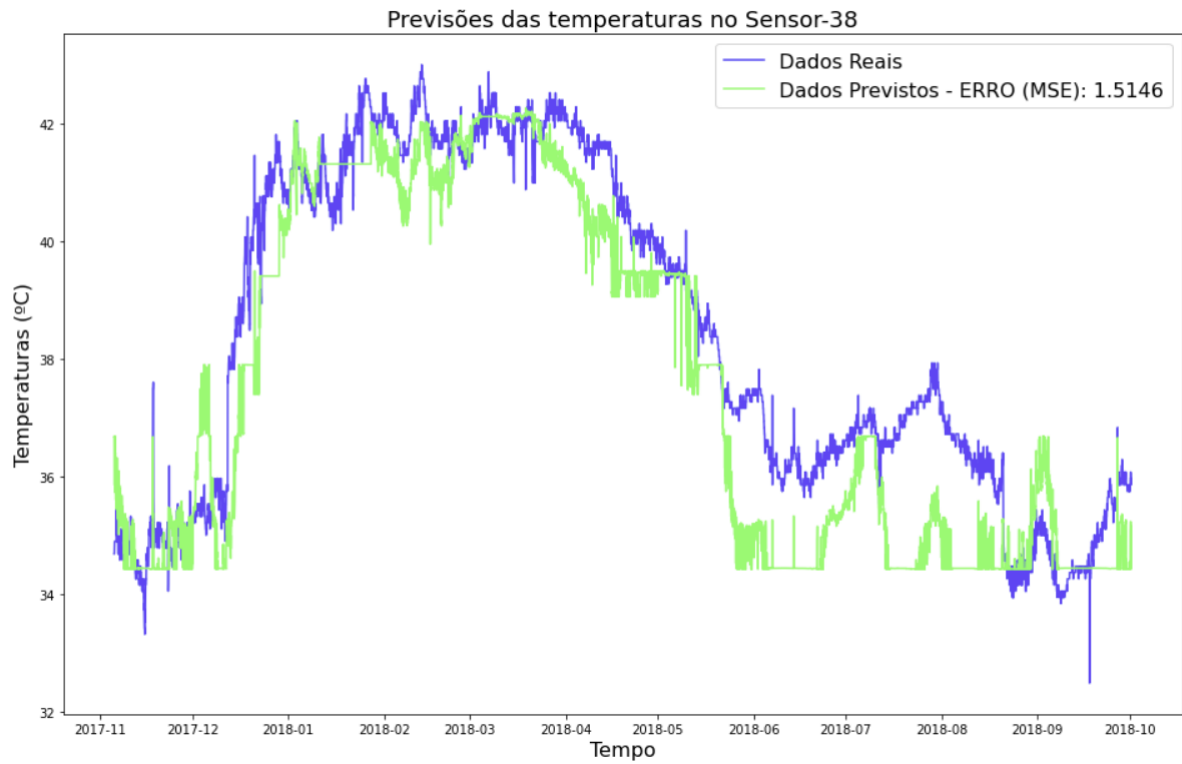


Figura 3.4: Previsão no ciclo 2017-2018 para o Sensor-38.

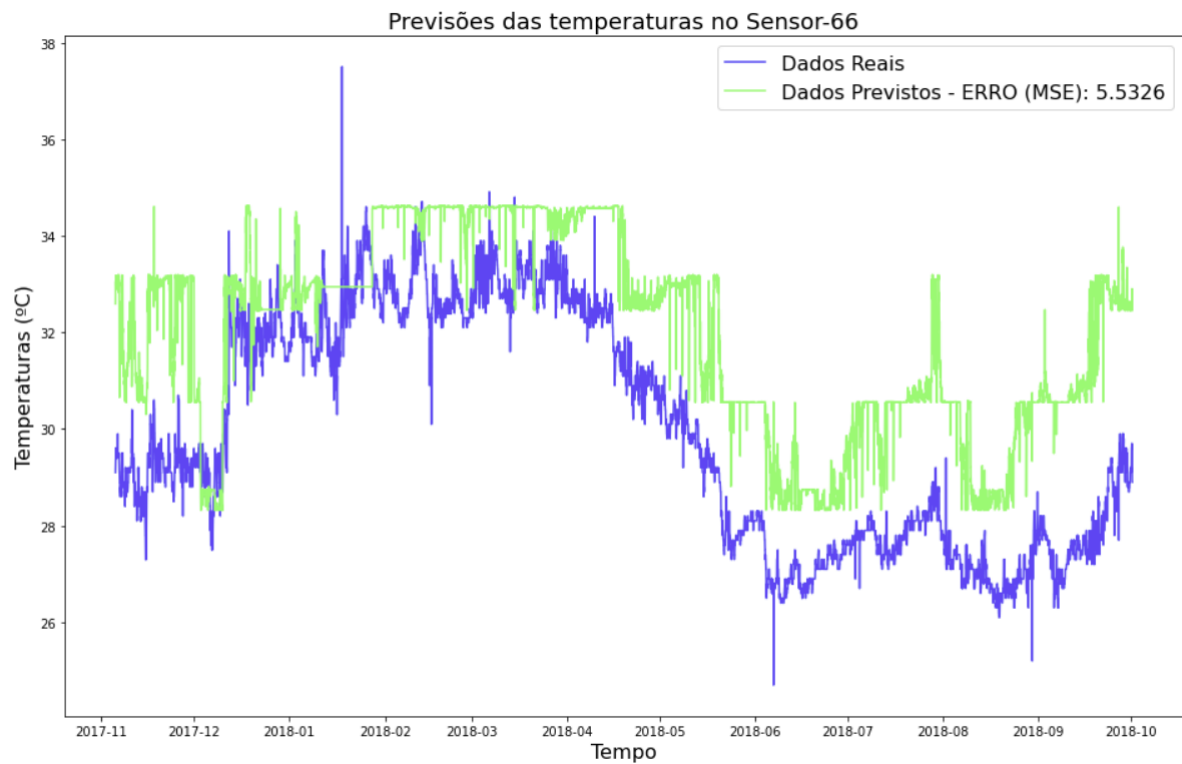


Figura 3.5: Previsão no ciclo 2017-2018 para o Sensor-66.

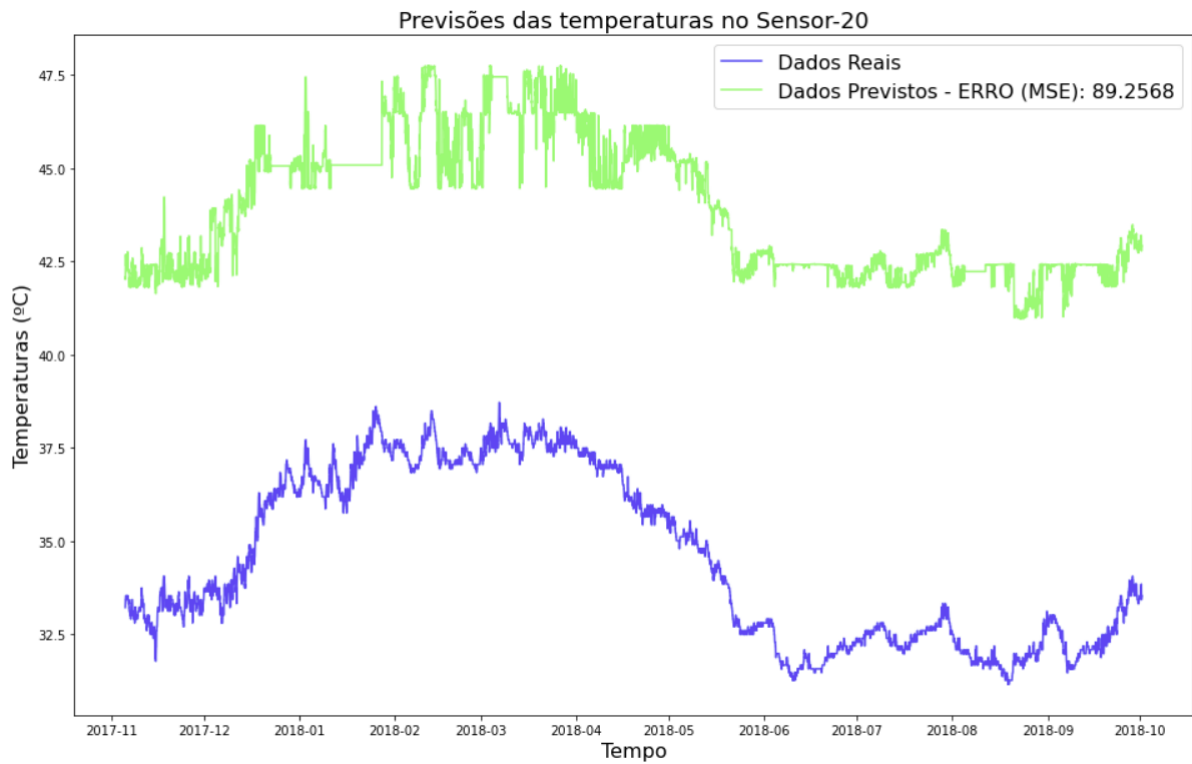


Figura 3.6: Previsão no ciclo 2017-2018 para o Sensor-20.

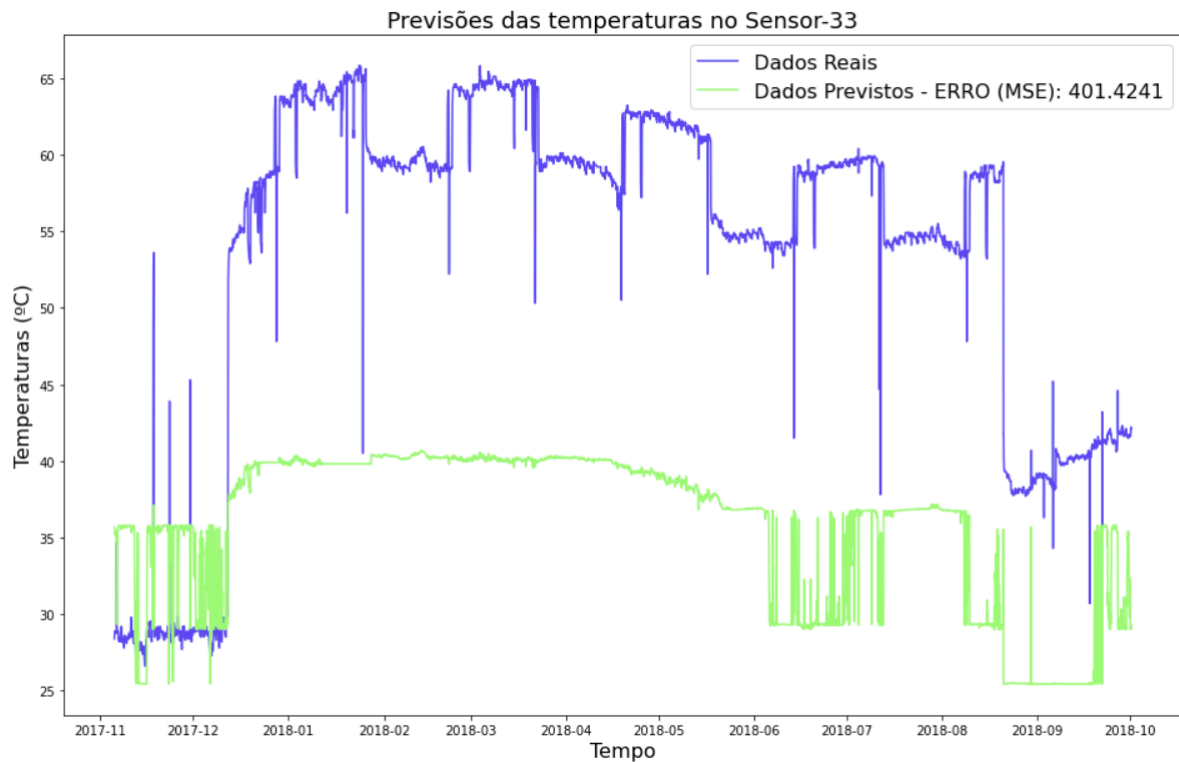


Figura 3.7: Previsão no ciclo 2017-2018 para o Sensor-33.

Pode-se observar pelos gráficos apresentados nas figuras 3.4, 3.5, 3.6 e 3.7 que:

- A previsão para alguns sensores foi boa, como no exemplo do Sensor-38, uma vez que o erro apresentado na previsão, se aproxima daquele encontrado no teste, igual a 1.0035, apresentado na tabela 3.2;
- Para alguns sensores, foi mediana, como no exemplo do Sensor-66. A curva prevista acompanha a curva real, porém o erro se tornou elevado, tendo em vista a média de treinamento apresentada na tabela 3.3 ser igual a 2.8701 e o erro de treinamento específico do Sensor-66 ter sido igual a 0.9353 como apresentado na tabela 3.2;
- Para outros sensores, como exemplificado pelas figuras 3.6 e 3.7, as previsões foram muito ruins.

A grande maioria das redes treinadas nessa etapa obteve uma previsão que ficou entre o que foi exemplificado pelo Sensor-38 e pelo Sensor-66. Porém os resultados bons e medianos vão no caminho de uma boa análise. O que chamou atenção foram as previsões encontradas em exemplos como o Sensor-20 e o Sensor-33, porque foram encontrados erros muito altos.

Em particular o caso do Sensor-20, nota-se que as curvas previstas e reais possuem o mesmo formato, porém há uma defasagem na faixa de 10 °C entre cada uma delas, quando o erro encontrado em teste para esse sensor foi de apenas 0.6963.

Para entender os resultados nesses casos em que esse comportamento inesperado ocorreu, voltou-se o trabalho à etapa de análise dos dados, em busca de compreender a relação entre os dados SICA-X e Sensor-X.

Para isso foram plotados em um mesmo gráfico os dados dos Sensores-X com seus respectivos SICA-X, conforme a seleção apresentada no item 3.1. Alguns desses gráficos são mostrados a seguir,

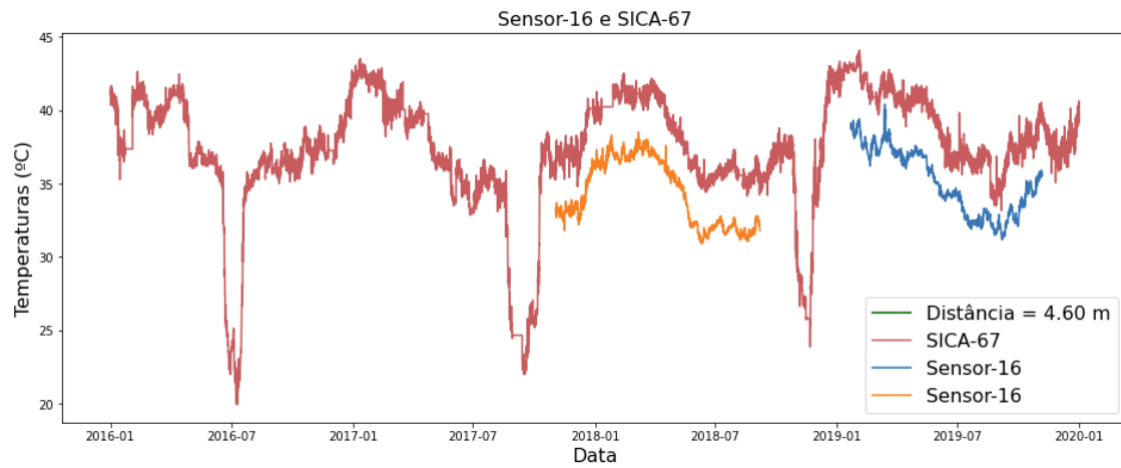


Figura 3.8: Comparação entre a distribuição de temperatura do Sensor-16 com o SICA-67.

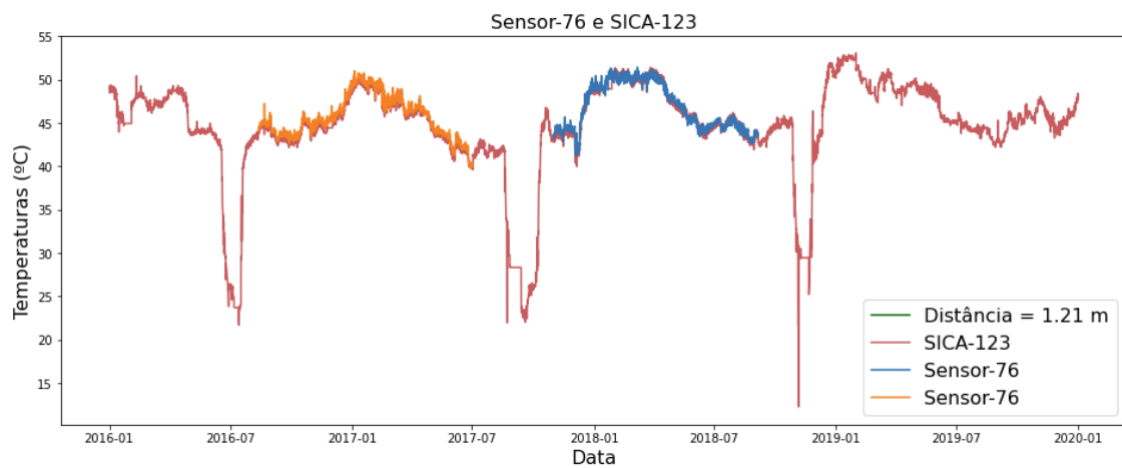


Figura 3.9: Comparação entre a distribuição de temperatura do Sensor-76 com o sensor SICA-123.

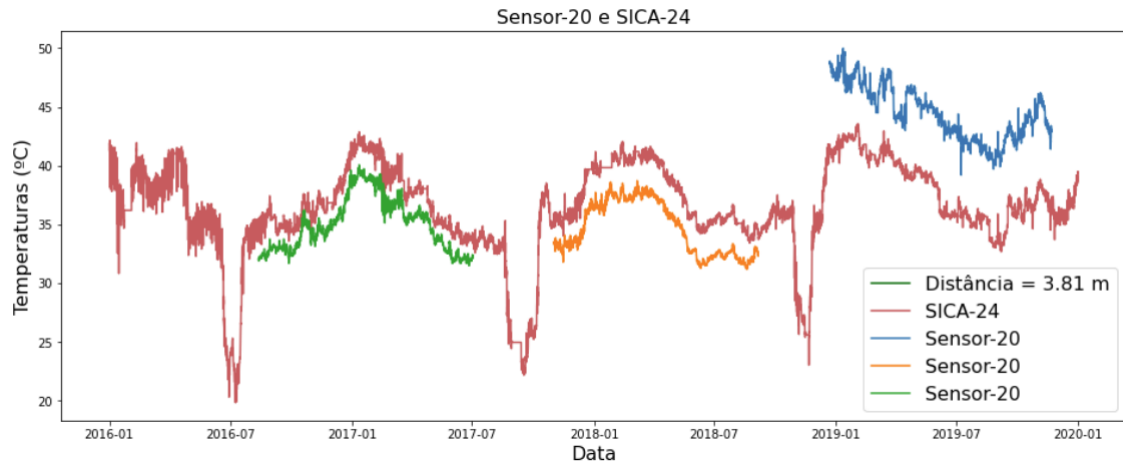


Figura 3.10: Comparação entre a distribuição de temperatura do Sensor-20 com o sensor SICA-24.

Observando visualmente a distribuição comparativa de temperaturas no tempo, entre os Sensores-X e os SICA-X, fica fácil supor que possuam uma boa correlação, uma vez que as curvas se comportam de forma similar, apesar de possíveis defasagens. Porém em algumas comparações, como exemplificado pelo gráfico da figura 3.10 nota-se que para alguns períodos o Sensor-X se comporta de uma forma em relação ao SICA-X, em outro período de outra forma.

Esse comportamento inesperado do Sensor-20, em particular, tendo no ciclo de operação de 2018-2019 ficado acima da curva do SICA-X e nos ciclos de operação anteriores tendo tido temperaturas inferiores à deste último faz surgir algumas hipóteses com relação a validação dos dados dos sensores-X:

- Sensor-20 foi trocado de lugar e essa troca não foi reportada;
- O equipamento físico do sensor-20 foi trocado e o novo equipamento passou a medir numa escala diferente;
- Descalibração do equipamento de um período para outro.

Não foi possível verificar as hipóteses, uma vez que a documentação dos Sensores-X não estava disponível, tendo disponível para esse trabalho apenas os dados referentes às medições. Dessa forma, foi necessário elaborar uma forma de correlacionar os dados do SICA-X com os dados dos Sensores-X, para evitar que falhas, como as exemplificadas pelo Sensor-20, chegassem a etapa de treinamento das redes neurais, o que faria com que as redes aprendessem uma informação inconsistente, impedindo-as de ter uma boa performance.

3.4 Modelagem Pós-Análise de Correlações

O passo seguinte, naturalmente, foi fazer um trabalho intermediário entre a seleção das variáveis e o treinamento, passo este que consistiu na análise de correlação entre os pares SICA x Sensor-X. Com esse passo, voltou-se à análise pré processamento com o objetivo de verificar as correlações entre os pares de sensores, utilizando métodos estatísticos, de forma a fazer um filtro em quais pares de variáveis possuem uma boa correlação, podendo assim ir para a etapa de treinamento e quais variáveis não se correlacionam bem, sendo essas últimas descartadas do treinamento.

3.4.1 Pré-Processamento

A segunda modelagem começou com um pré-processamento mais elaborado. Como mencionado no parágrafo anterior, métodos estatísticos foram utilizados para medir a correlação entre os pares de variáveis que iriam para o treinamento.

Um dos métodos utilizados foi a correlação de Spearman, apresentado no item 2.4.2, um método estatístico de correlacionar variáveis, que independe que elas sejam linearmente dependentes, o que se mostrou bastante conveniente para o problema em questão, uma vez que não há uma dependência direta entre as medições de temperatura consideradas.

Foi utilizado também um segundo método, em parte desenvolvido neste trabalho, que consistiu em calcular a diferença (nomeada de delta, no escopo deste trabalho) ponto a ponto entre a temperatura do Sensor-X e seu par SICA-X, imaginando que quanto mais linear for a distribuição desse delta no tempo, maior a correlação entre as variáveis, uma vez que um delta distribuído no tempo constante representaria que as temperaturas do sensor previsto e do sensor preditor possuem a mesma distribuição, separadas apenas por uma defasagem igual ao valor de delta.

Para a análise da distribuição dos deltas, foi utilizada a medida de kurtosis, apresentada no item 2.8, uma medida de altura de distribuições gaussianas, onde um valor de kurtosis igual a zero implica numa altura perfeita em relação a distribuição normal, valores menores do que zero indicam uma curva com altura menor e valores positivos indicam a altura da curva acima da normal.

O interesse no problema em questão é que a distribuição dos deltas seja o mais próxima da média possível, o que implica numa curva mais alta, então valores mais altos de kurtosis são os ideais para o escopo deste trabalho.

Os gráficos utilizados nessa etapa de análise foram os mesmos exemplificados pelas figuras 3.8, 3.9 e 3.10, além dos gráficos das distribuições delta e a distribuição de frequência de suas temperaturas, como exemplificados a seguir,

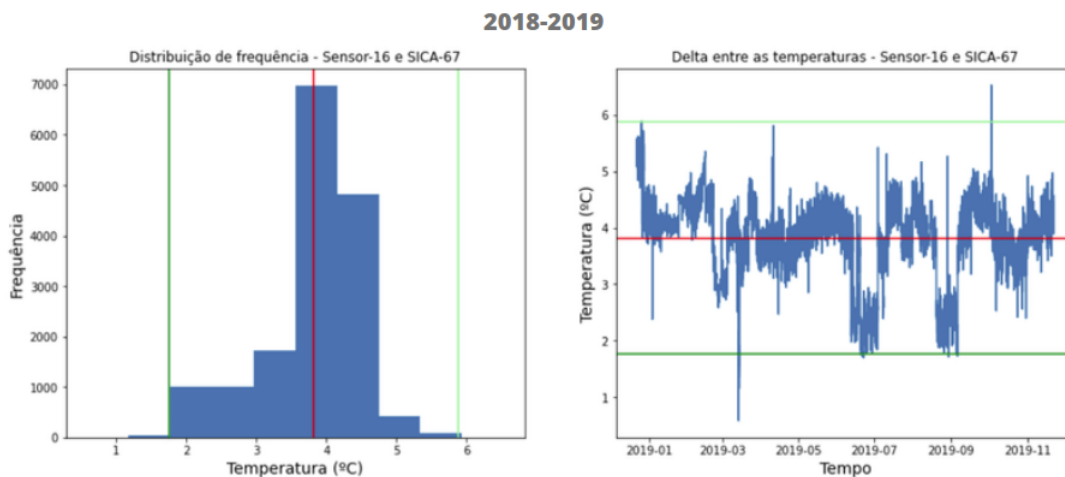


Figura 3.11: Para 2018-2019. À Esquerda: distribuição de Frequência de Temperaturas do Delta entre o Sensor-16 e o SICA-67. À Direita: distribuição do Delta no tempo.

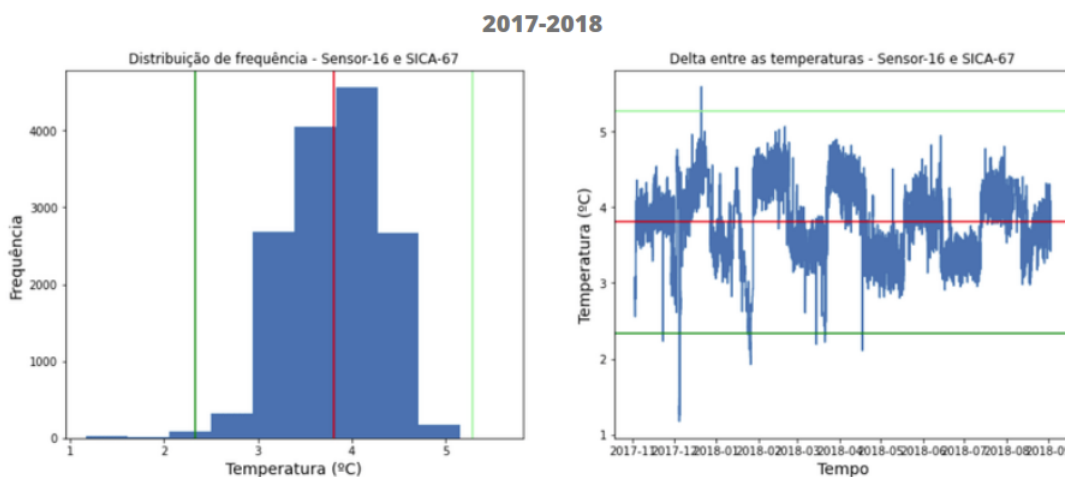


Figura 3.12: Para 2017-2018. À Esquerda: distribuição de Frequência de Temperaturas do Delta entre o Sensor-16 e o SICA-67. À Direita: distribuição do Delta no tempo.

Nas figuras 3.11 e 3.12, os gráficos à esquerda, representando a distribuição de frequência pode ser utilizado para uma análise visual em relação a distribuição normal, que é base para o cálculo da kurtosis da distribuição. Os gráficos à direita mostram a distribuição de temperatura dos deltas, junto com as marcações de ± 3

desvios padrão.

Com os gráficos implementados, e os métodos de kurtosis e Spearman escolhidos, foi necessária a escolha da forma como eles seriam utilizados e essa forma foi particular para cada par de sensores, dependendo das características.

Alguns sensores-X possuem medições em mais de um ciclo de operação, onde para cada ciclo de operação, a correlação entre essa variável e a variável preditora é diferente. Dessa forma, foi escolhido que,

- Para os sensores onde há medições em mais de um período: Caso apenas um dos métodos de correlação utilizados fosse satisfatório em cada um dos períodos, o sensor estaria apto a ir para o treinamento da rede neural;
- Para sensores onde só um período possui medição: Ambos os métodos de correlação precisariam ser cumpridos para levar o sensor para o treinamento.

Com essa forma de avaliar em mãos, foram escolhidos os valores limites de 0.5 para Spearman, onde valores iguais ou maiores seriam aceitos e de 2 para kurtosis, onde também, valores iguais ou maiores seriam aceitos. Então foram calculadas a Spearman, o delta e a kurtosis do delta para cada par de sensores e através de uma análise visual de cada valor e dos gráficos de cada distribuição no tempo, foram selecionados os sensores que iriam para o treinamento e os que seriam descartados por não apresentarem boa correlação.

Definidos os limites para cada um dos métodos de avaliação e definido que visualmente, não se poderia aceitar correlações inconsistentes, como aquela apresentada pelo Sensor-20, exemplificado na figura 3.10, foram então feitos os cálculos das métricas de avaliação para cada par de sensores em cada ciclo de operação disponível, e então feita a seleção de quais pares de sensores não seriam utilizados e quais iriam para a próxima etapa, para serem dados de treino das redes neurais.

Capítulo 4

Resultados

Utilizando a modelagem apresentada no capítulo anterior e seguindo ao passo intermediário de análise de correlação e filtragem dos dados com boa correlação, que possui crucial importância para a qualidade dos métodos propostos, voltou-se o trabalho para a modelagem e treinamento, em versão final, das redes neurais.

Seguindo a ordem natural de elaboração dos modelos, o modelo mais simples foi o primeiro a ser treinado, sendo assim o modelo DNN.

4.1 Treinamento DNN

A mesma abordagem do capítulo anterior foi utilizada, separando os dados em treino, validação e teste, exatamente como descrito no item 3.3.1 e então vários modelos foram treinados com diferentes hiperparâmetros.

4.1.1 Resultados DNN

Após o treinamento de modelos com diversos hiperparâmetros, o modelo que apresentou o melhor resultado, avaliado pelo erro médio em teste, foi o modelo cujos hiperparâmetros são apresentados na tabela 4.1.

Tabela 4.1: Parâmetros Utilizados no melhor modelo de DNN, após análise das correlações.

Parâmetro	Valor
Número de Épocas	400
Número de Camadas Internas	7
Número de Neurônios por Camada	[128, 512, 128, 256, 128, 512, 128]

Na primeira etapa (item 3.3.1) foram treinadas 36 redes para cada modelo, porém como foram feitos filtros para selecionar apenas os pares de sensores com boa correlação, 11 pares de sensores foram removidos. Dessa forma, para esta etapa, foram treinadas 25 redes para cada modelos e os resultados de cada uma das 25, para o melhor modelo, são apresentados na tabela a 4.2.

Tabela 4.2: Erros do Modelo DNN após análise de correlação.

Sensor	Erro em Treinamento	Erro em Validação	Erro em Teste
Sensor-03	0.8232	0.8615	0.8283
Sensor-07	2.3662	2.3143	2.3184
Sensor-16	1.6170	1.6481	1.6226
Sensor-18	1.5656	1.5465	1.5266
Sensor-22	0.2233	0.2271	0.2146
Sensor-23	0.5085	0.5225	0.4980
Sensor-24	0.5560	0.5147	0.5274
Sensor-28	1.4212	1.4020	1.4355
Sensor-29	1.3395	1.3279	1.3351
Sensor-30	1.1439	1.1905	1.1356
Sensor-41	5.6421	5.2942	5.5636
Sensor-51	11.7795	12.0542	11.9465
Sensor-52	0.9240	0.9306	0.9216
Sensor-54	0.7805	0.7828	0.7831
Sensor-58	0.5362	0.5193	0.5351
Sensor-59	0.4402	0.4371	0.4410
Sensor-60	0.5625	0.5469	0.5207
Sensor-63	1.0221	1.0538	1.0774
Sensor-66	0.8302	0.8026	0.8640

Continua na próxima página

Tabela 4.2 – Erros do Modelo DNN após análise de correlação

Sensor	Erro em Treinamento	Erro em Validação	Erro em Teste
Sensor-67	1.4689	1.3928	1.4811
Sensor-72	0.6757	0.6874	0.6720
Sensor-74	0.2937	0.2977	0.2998
Sensor-75	0.3520	0.3343	0.3574
Sensor-78	0.7863	0.8015	0.7939
Sensor-82	3.6989	3.6285	3.5651

As estatísticas referentes a esse modelo, são apresentadas na tabela 4.3

Tabela 4.3: Estatísticas do Modelo DNN após análise de correlação.

	Erro em Treinamento	Erro em Validação	Erro em Teste
Contagem	25	25	25
Média	1.6543	1.6447	1.6518
Desvio Padrão	2.4185	2.4415	2.4430
Valor Mínimo	0.2233	0.2270	0.2146
Valor Máximo	11.7795	12.0542	11.9765

Observando os dados de treinamento e teste, nota-se que a média do erro em teste foi cerca de 42% menor no modelo após a análise das correlações, em relação àquele feito previamente, cujas estatísticas são apresentadas na tabela 3.3. Vale observar também que o desvio padrão também foi menor no novo modelo, porém os valores mínimos e máximo foram maiores, dessa forma, o melhor valor pontual foi encontrado no modelo anterior e o pior valor pontual, no novo modelo. Entretanto, o objetivo de um modelo de rede neural é que ele seja capaz de generalizar o aprendizado, dessa forma o modelo com melhor média é mais interessante do que um modelo com algumas melhores avaliações pontuais, e nesse ponto o modelo novo foi capaz de melhorar a média de erro em 42% na etapa de treinamento e teste.

4.2 Treinamento LSTM

Com o melhor modelo de DNN treinado, o passo seguinte naturalmente foi partir para a arquitetura LSTM, onde as maiores expectativas de melhores resultados foram postas, por ser uma arquitetura recorrente, que por essa característica

intrínseca espera-se melhores resultados em séries temporais.

A preparação para o treinamento da LSTM foi a mesma que a utilizada na DNN, já com os dados filtrados pelas correlações e com os 25 pares de sensores selecionados. Foram feitas as mesmas etapas de divisão em dados de treino, validação e teste, seguindo a mesma proporção de 64%, 16% e 20%, respectivamente.

Uma característica que diferencia da LSTM para a DNN na modelagem de sua arquitetura é o fato de por ser uma rede neural recorrente, ela poder receber uma série de valores como entrada e não apenas um valor individual, como acontece no caso das redes neurais não recorrentes.

Diante dessa característica das redes recorrentes, foram escolhidas séries de entradas para a LSTM com os intervalos descritos na tabela 4.4.

Tabela 4.4: Séries de valores de entrada no treinamento da LSTM.

Tamanho da Série	Número de pontos
30 minutos	1 ponto
1 dia	48 pontos
2 dias	96 pontos
7 dias	336 pontos
15 dias	720 pontos
30 dias	1440 pontos
60 dias	2880 pontos

É importante observar que apesar de ter sido representada na tabela 4.4 como uma série de valores, as redes treinadas com a série de 30 minutos na verdade foram treinadas ponto a ponto, da mesma forma como a DNN, uma vez que os dados são apresentados em intervalos de 30 minutos. Esse treinamento foi feito como forma de comparar a performance da LSTM em um treinamento ponto a ponto com a performance dela mesma recebendo como entrada uma série de valores, bem como comparar com a DNN ponto a ponto.

Os mesmos hiperparâmetros foram alterados, e novamente diversos modelos foram treinados e testados, para cada uma das séries de entradas, tendo o que apresentou o melhor resultado sido escolhido para a análise final e para a comparação com o modelo DNN treinado anteriormente.

4.2.1 Resultados LSTM

Os hiperparâmetros do modelo melhor avaliado de LSTM estão apresentados na tabela 4.5.

Tabela 4.5: Parâmetros Utilizados no melhor modelo de LSTM, após análise das correlações.

Parâmetro	Valor
Número de Épocas	800
Tamanho da série de Entrada	30 minutos
Número de Camadas Internas	7
Número de Células por Camada	[256, 512, 256, 512, 256, 512, 256]

É curioso notar que o melhor resultado da LSTM foi também aquela onde foi treinada com a entrada de apenas um ponto, assim como foi treinada a DNN. Esse fato é contraintuitivo, uma vez que esperasse que a LSTM tenha um resultado melhor quando sua característica de recorrência está em ação. Na sequência da apresentação dos resultados, será apresentada uma discussão sobre possíveis hipóteses para esse comportamento.

Os resultados individuais para cada um dos 25 sensores são apresentados na tabela 4.6.

Tabela 4.6: Erros do Modelo LSTM após análise de correlação.

Sensor	Erro em Treinamento	Erro em Validação	Erro em Teste
Sensor-03	0.9273	0.9412	0.9590
Sensor-07	1.1886	1.1594	1.1873
Sensor-16	0.4688	0.4657	0.4734
Sensor-18	0.1104	0.1107	0.1142
Sensor-22	0.1015	0.0936	0.0917
Sensor-23	0.1129	0.1164	0.1119
Sensor-24	0.1119	0.1244	0.1295
Sensor-28	0.0224	0.0211	0.0226
Sensor-29	0.0883	0.0886	0.0876
Sensor-30	0.0804	0.0780	0.0804
Sensor-41	6.6340	6.7651	6.9204
Sensor-51	10.2988	10.4057	10.2264
Sensor-52	1.2280	1.0653	1.1767
Sensor-54	0.7467	0.7620	0.7329
Sensor-58	0.3917	0.4147	0.3704
Sensor-59	0.3351	0.3432	0.3398
Sensor-60	0.3893	0.3992	0.4140
Sensor-63	0.1160	0.1142	0.1161
Sensor-66	0.7802	0.7498	0.7634
Sensor-67	0.9118	0.8819	0.9031
Sensor-72	0.4892	0.5573	0.4978
Sensor-74	0.1652	0.1572	0.1609
Sensor-75	0.1537	0.1500	0.1523
Sensor-78	0.1967	0.1963	0.2123
Sensor-82	4.3016	4.3798	4.3056

As estatísticas referentes a esse modelo, são apresentadas na tabela 4.7

Tabela 4.7: Estatísticas do Modelo LSTM após análise de correlação.

	Erro em Treinamento	Erro em Validação	Erro em Teste
Contagem	25	25	25
Média	1.2144	1.2216	1.2207
Desvio Padrão	2.4025	2.4353	2.4190
Valor Mínimo	0.0224	0.0211	0.0226
Valor Máximo	10.2988	10.4057	10.2264

Através de uma avaliação da tabela 4.7, pode-se notar uma melhora no erro médio de teste de 57.47% em relação ao modelo de DNN pré-análise de correlações e de 26.09% em relação ao modelo de DNN pós-análise de correlações. Além do erro médio, o desvio padrão também se mostrou ainda menor, porém ainda com um valor alto, o que é justificado devido à uma medida pontual ter tido um erro muito alto. O erro pontual mínimo na LSTM foi extremamente baixo, sendo o menor dentro os 3 modelos.

Avaliando apenas por treinamento, validação e teste, a LSTM foi a arquitetura que entregou melhores resultados, porém, como dito no item 3.3.1, é necessária ainda uma avaliação dos resultados em uma inferência em dados de um ciclo de operação diferente daquele utilizado na etapa de treinamento. Como foi feito no item 3.3.1, foi utilizado o ciclo de operação 2017-2018 para essa inferência e avaliação.

Foram plotados gráficos comparando os dados reais dos Sensores-X, com os dados das previsões feitas para esses sensores tanto pela DNN pós-análise de correlação, quanto pela LSTM, também pós-análise de correlação. Alguns desses gráficos são apresentados a seguir,

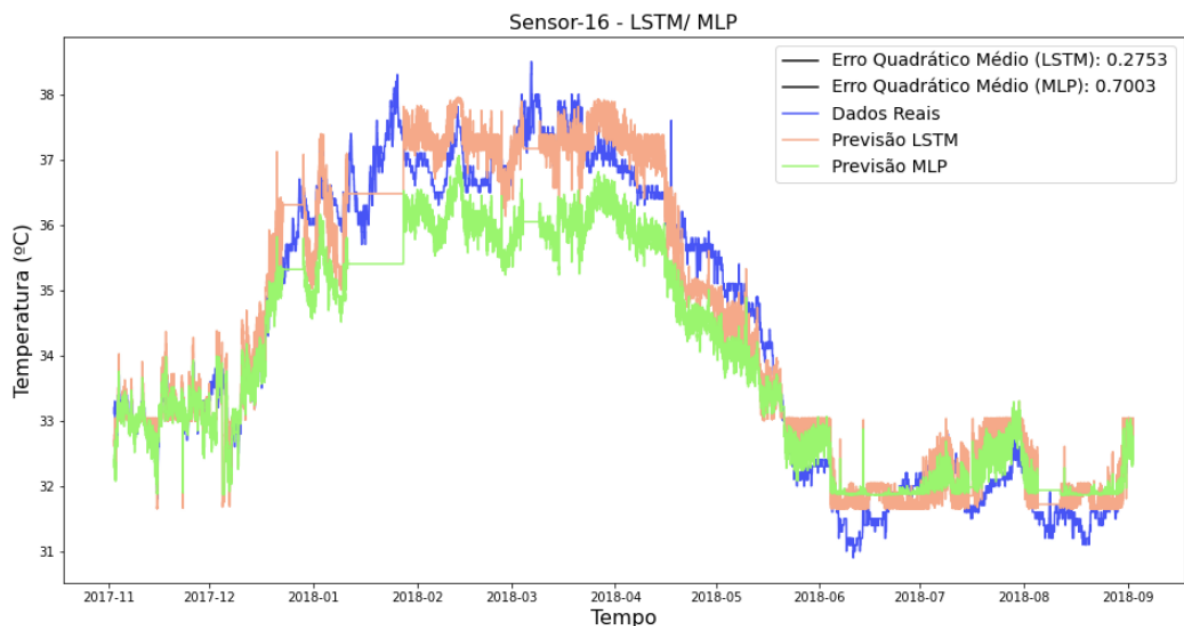


Figura 4.1: Comparação entre o valor real para o Sensor-16 e suas previsões pela LSTM e pela DNN.

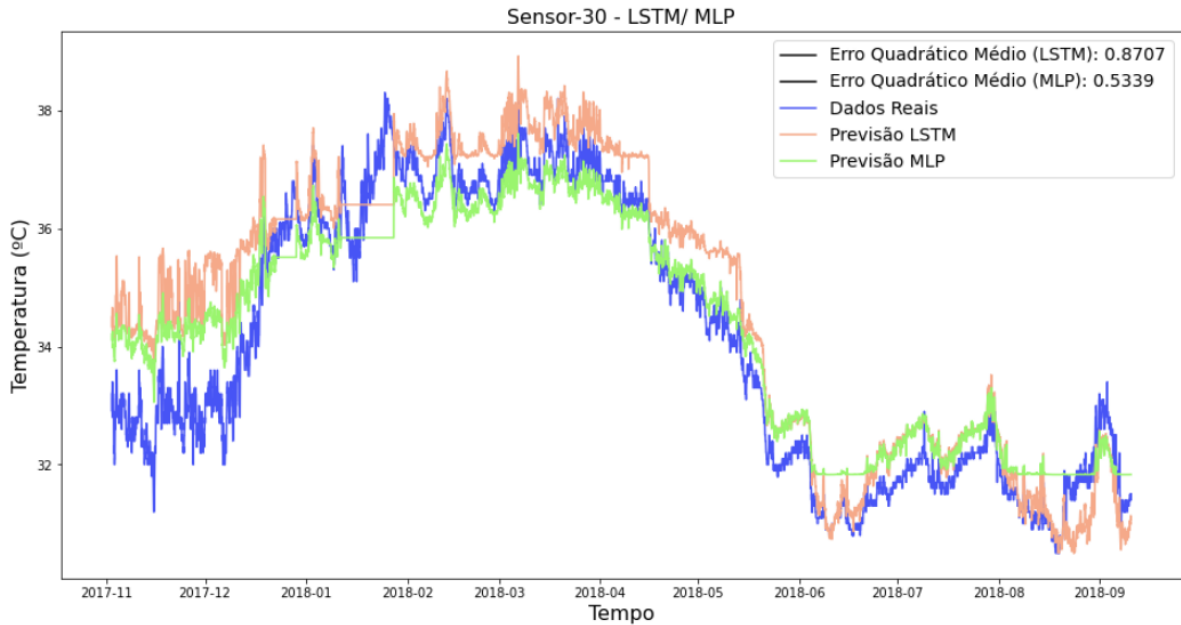


Figura 4.2: Comparação entre o valor real para o Sensor-30 e suas previsões pela LSTM e pela DNN.

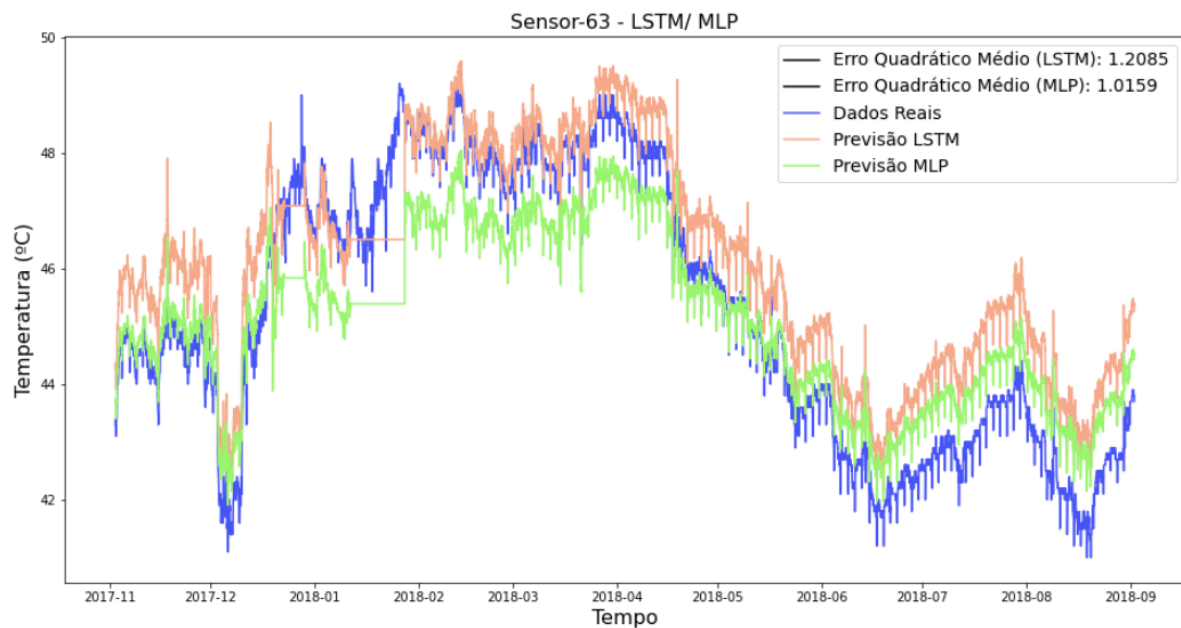


Figura 4.3: Comparação entre o valor real para o Sensor-63 e suas previsões pela LSTM e pela DNN.

Na tabela 4.8 , são apresentados os erros médios quadráticos para cada um dos sensores nos quais foi feita a inferência no ciclo de operação 2017-2018, tanto para a LSTM, quanto para a DNN.

Tabela 4.8: Erro quadrático médio na inferência para o ciclo de operação 2017-2018.

Sensor	Erro Quadrático Médio LSTM	Erro Quadrático Médio DNN
Sensor-16	0.2753	0.7003
Sensor-18	0.2565	0.6933
Sensor-23	2.9102	2.5645
Sensor-24	2.3393	2.0547
Sensor-28	0.4773	1.2419
Sensor-29	0.9997	0.6581
Sensor-30	0.8707	0.5339
Sensor-41	4.3413	4.2395
Sensor-52	5.5885	5.4157
Sensor-54	1.3375	1.3455
Sensor-58	4.6682	4.2772
Sensor-59	4.5186	4.4706
Sensor-60	4.4661	4.4716
Sensor-63	1.2085	1.0159
Sensor-66	5.1682	5.3289
Sensor-67	3.2944	4.5154
Sensor-72	2.6812	2.9476
Sensor-82	8.1902	8.4343

As estatísticas desses resultados são apresentadas na tabela 4.9,

Tabela 4.9: Estatísticas das previsões.

	Erros LSTM	Erros DNN
Contagem	18	18
Média	2.9773	3.0505
Desvio Padrão	2.2205	2.2171
Valor Mínimo	0.2565	0.5339
Valor Máximo	8.1902	8.4334

A tabela 4.9 compara o desempenho do melhor modelo de LSTM com o melhor modelo de DNN treinado nesse trabalho. Apesar de na etapa de treinamento e teste em um ciclo de operação a LSTM ter obtido resultados melhores, na hora de entregar dados de um ciclo de operação diferente, já pensando na etapa de inferência das temperaturas, ambas as arquiteturas apresentaram resultados muito próximos, com a LSTM apresentando um erro 2.4% menor.

Os resultados finais trazem uma informação esperada, de que as redes são capazes de serem generalizadas para aprender a relação entre os dados, porém era esperado também que a LSTM obtivesse uma resposta muito superior à DNN, o que não se mostrou verdadeiro.

Duas hipóteses podem ser levantadas a respeito do resultado inesperado da LSTM, no sentido de não se mostrar muito superior à DNN na fase de inferência de temperatura em um ciclo diferente daquele no qual as redes foram treinadas,

- Hipótese 1: Os treinamentos foram feitos com poucos dados;
- Hipótese 2: O intervalo de um ano não representou uma janela de tempo suficiente para que um comportamento de recorrência acontecesse.

A primeira hipótese é uma verdade, as redes foram sim treinadas com poucos dados. Apesar de cada sensor treinado possuir dezenas de milhares de registros, esse número é relativamente baixo para um treinamento de rede neural e junto desse fato, foram treinadas apenas 25 redes por modelo, após o filtragem das correlações entre Sensor-X vs SICA-X. Porém a quantidade de dados influencia em ambas arquiteturas. Com mais dados, muito possivelmente tanto a LSTM quanto a DNN teriam melhores desempenhos, porém não se sabe se a melhora no desempenho da LSTM seria expressivamente superior à da DNN.

A hipótese 2, apesar de estar de certa forma relacionada com a primeira, na questão da quantidade de dados, toca num ponto relevante especificamente para as arquiteturas de redes neurais recorrentes, uma vez que para a rede aprender uma recorrência, a recorrência deve primeiro existir nos dados. Uma vez que os dados que são passados para o treinamento representam um ciclo completo de operação da usina, pode-se imaginar que não há uma condição de forte recorrência nessa janela temporal do ciclo de operação, fazendo com que a rede neural não tenha sua melhor eficiência no treinamento recorrente.

A hipótese 2 ganha mais força quando se observa o fato de o melhor resultado da LSTM ser aquele onde os dados de entrada não estavam sendo passados na forma de uma série e sim de valores individuais, ou seja, a LSTM estava tratando ponto a ponto, da mesma forma como a DNN foi treinada anteriormente.

Além dessas hipóteses, em (PINHEIRO, 2020), o autor levanta uma hipótese diferente, a cerca da quantidade de ruídos nos dados. Redes neurais recorrentes são

muito boas em treinar dados sequenciais, porém desde que não possuam muitos ruídos. Essa hipótese também pode ser uma justificativa para o desempenho semelhante de ambas arquiteturas apresentadas no presente trabalho, uma vez que os dados de temperatura utilizados apresentam uma quantidade relativamente grande de ruídos, algo intrínseco a medições reais.

Pela tabela 4.9 tem-se que a média de erros para as previsões em um ciclo de operação diferente daquele onde as redes neurais foram treinadas e testadas está em torno de 3, como a métrica de medição é o erro quadrático médio, então essa medida está numa unidade de C^2 , fazendo uma raiz quadrada desses erros, tem-se que a média de erro absoluto fica em torno de $\pm 1.72^\circ\text{C}$. Muitos sensores de temperatura projetados para medir temperaturas acima de 100°C , possuem precisão de 1°C , fazendo essa análise nota-se que a média de erro absoluto dos modelos de redes neurais muito se aproxima da própria incerteza do medidor de temperatura, fazendo com que o resultado possa ser interpretado como satisfatório e indicando que o método de utilização de redes neurais na inferência das temperaturas para o passado é promissor.

Em se tratando da comparação das arquiteturas LSTM e DNN, no presente trabalho obteve-se uma performance semelhante, porém ligeiramente superior da arquitetura LSTM. Em (PINHEIRO, 2020), o autor também fazendo um estudo sobre inferência de temperaturas, encontrou uma performance ligeiramente superior para a arquitetura DNN. Em (SHAH *et al.*, 2018), também fazendo uma comparação entre as arquiteturas mencionadas, porém num problema de previsão de preço no mercado de ações, o autor encontrou resultados bons para ambas arquiteturas, com um pouco de superioridade para a arquitetura recorrente. Em (SHEWALKAR, 2018), num estudo sobre reconhecimento de voz, também fazendo uma comparação de diferentes arquiteturas, o autor encontrou melhores resultados também para a LSTM.

Observando os resultados de diferentes comparações em diferentes conjuntos de dados na literatura, pode-se notar que de modo geral a LSTM tem uma melhor performance em problemas de dados sequenciais, como previsão de séries temporais e reconhecimento de áudios, como exemplificados pelos estudos citados.

Capítulo 5

Conclusão

O objetivo deste trabalho foi estudar e desenvolver uma solução para o problema do estudo de envelhecimento dos equipamentos elétricos e eletrônicos da usina nuclear Angra 1. O envelhecimento dos equipamentos é de extrema importância para a elaboração do programa de qualificação ambiental, que faz parte do programa de extensão de vida útil.

Angra 1 ao longo de seus anos de operação, não contou com medidores de temperatura estrategicamente localizados para um estudo de envelhecimento dos equipamentos, sendo assim inviável fazer esse estudo de forma precisa e direta. A partir do ano de 2015, com os primeiros passos sendo dados em direção à extensão da vida útil da planta, começaram a ser instalados medidores de temperatura em posições favoráveis ao estudo de envelhecimento e a partir dessas instalações nasce o propósito deste trabalho.

A partir da instalação dos medidores de temperatura em posições favoráveis, foi possível desenvolver um modelo de redes neurais artificiais capaz de receber como entradas as medições de temperatura dos sensores do Sistema de Computadores de Angra (SICA) e chegar nas saídas dos medidores instalados. Com um modelo capaz de criar um histórico de temperaturas a partir da inferência das temperaturas do SICA passa a ser possível fazer o estudo do envelhecimento dos equipamentos de segurança, necessário ao Programa de Qualificação Ambiental, viabilizando a extensão de vida útil da usina.

Para simplificar o entendimento da modelagem, define-se como SICA-X, um nome hipotético para um sensor do Sistema de Computadores, onde X representa um número e esse nome hipotético representa um sensor real do SICA. O mesmo se aplica aos sensores instalados a partir do ano de 2015, onde foi dado o nome hipotético de Sensor-X, onde X representa um número e esse nome hipotético

representa um sensor real instalado.

A modelagem proposta foi relativamente simples, onde um sensor SICA-X é passado para o modelo e esse modelo deve retornar uma medições que generalize aquela apresentada por um Sensor-X. Na etapa de treinamento, os dados de SICA-X são passados ao modelo e ele deve aprender que a saída deve ser a mais próxima possível do valor de Sensor-X.

A partir dessa modelagem, foi feita uma seleção de variáveis, para encontrar os pares de Sensor-X e SICA-X que apresentassem melhor correlação entre si, para que o modelo fosse treinado com pares de boas correlações.

Dada a modelagem, a tecnologia escolhida para fazer esse aprendizado e a inferência, foi a tecnologia das redes neurais artificiais, em particular duas de suas arquiteturas. A primeira arquitetura a ser modelada foi a DNN (*Deep Neural Network*), a mais simples das arquiteturas de redes neurais profundas. A segunda arquitetura foi a LSTM (*Long-Short Term Memory*), uma arquitetura recorrente, extremamente robusta para problemas sequenciais.

Diversos modelos com diferentes hiperparâmetros foram treinados e testados em busca de modelos que apresentassem os melhores resultados. Durante a etapa de avaliação dos resultados do primeiro modelo de DNN, um problema foi encontrado a cerca da correlação entre as variáveis SICA-X e Sensor-X e um passo intermediário precisou ser elaborado.

Nesse passo intermediário foram usados métodos estatísticos, como a correlação de Spearman e a Kurtosis da distribuição normal, para avaliar a qualidade da correlação entre as variáveis de entrada e de saída das redes neurais. Com essa correlação estatística foram filtrados alguns pares de sensores que não apresentaram boas métricas de correlação, para que as redes fossem treinadas apenas com variáveis que se correlacionassem muito bem.

Após essa etapa intermediária os treinamentos voltaram a ser feitos e os resultados se mostraram muito superiores tanto com as novas DNNs quanto com a arquitetura LSTM.

A métrica utilizada na avaliação dos modelos de redes neurais foi o erro médio quadrático e na avaliação final, os melhores modelos encontraram avaliações de 2.9773 e 3.0505 para a média dos erros médios quadráticos, para a LSTM e para a

DNN respectivamente.

Como a métrica é um valor quadrático, fazer a raiz quadrada dos resultados dá uma melhor dimensão de análise, pois transforma os valores na unidade de temperatura em questão. Sendo assim, os erros médio absolutos encontrados foram de 1.72 e 1.75 para a LSTM e para a DNN, respectivamente. Esses erros indicam que as medições previstas estão numa faixa de aproximadamente 1.7°C de incerteza das medições reais, em média.

Fazendo um paralelo com o pensamento de que em alguns pontos do interior do prédio de contenção do reator da usina, as temperaturas podem chegar a valores superiores as 100°C, os medidores de temperatura instalados precisam ser capazes de medir temperaturas elevadas e para medidores para essas dimensões, a incerteza da medição costuma ser em torno de 1°C. Analisando do ponto de vista de que a incerteza dos medidores reais é de 1°C, uma imprecisão de aproximadamente 1.7°C, inferida por um modelo de rede neural, é uma imprecisão muito boa, o que torna o modelo eficaz na inferência das temperaturas em todos os ciclos de operação, a fim de criar um histórico de temperatura viabilizando o estudo de envelhecimento dos componentes dos sistemas de segurança da usina.

Trabalhos Futuros

- Estudos do mesmo problema com arquiteturas de redes neurais diferentes das utilizadas nesse trabalho.
- Aplicação de redes neurais diretamente no cálculo do envelhecimento dos equipamentos, utilizando a equação de Arrhenius como modelo de treinamento.

Referências Bibliográficas

- [1] AN YOSHUA BENGIO, X. G., 2010, “Understanding the difficulty of training deep feedforward neural networks”, .
- [2] ARCHDEACON, T. J., 1994, *Correlation and Regression Analysis: A Historian’s Guide*. Univ of Wisconsin Press.
- [3] BENESTY, J., CHEN, J., HUANG, Y., 2008, “On the Importance of the Pearson Correlation Coefficient in Noise Reduction”, *IEEE Transactions on Audio, Speech, and Language Processing*, v. 16, n. 4, pp. 757–765. doi: 10.1109/TASL.2008.919072.
- [4] CARDOZO, F. H. P., 2019, “Simulação de Detectores de Radiação no Interior do Prédio de Contenção da Usina Nuclear Angra 1”, .
- [5] CAUCHY, A., 1847, “Méthode générale pour la résolution des systèmes d’équations simultanées”, .
- [6] CHEN, X., ISHWARAN, H., 2012, “Random forests for genomic data analysis”, *Genomics*, v. 99, n. 6, pp. 323–329. ISSN: 0888-7543. doi: <https://doi.org/10.1016/j.ygeno.2012.04.003>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0888754312000626>>.
- [7] CNEN, 1984, “CNEN NE 1.04, Licenciamento de Instalações Nucleares”, .
- [8] DAVID, P. K.; MONTANARI, G. C., 1992, “Compensation effect in thermal aging investigated according to Eyring and Arrhenius models”, .
- [9] DE CARVALHO, H. V. F., CARVALHO, E. C., ARRUDA, H., et al., 2018, “Detecção de Anomalias em Comportamento de Abelhas Utilizando Redes Neurais Recorrentes”. In: *Anais do IX Workshop de Computação Aplicada a Gestão do Meio Ambiente e Recursos Naturais*, Porto Alegre, RS, Brasil. SBC. doi: 10.5753/wcama.2018.2931. Disponível em: <<https://sol.sbc.org.br/index.php/wcama/article/view/2931>>.
- [10] DE SOUZA, D., 2019, “Spotlight marca nova etapa no setor nuclear, com extensão da vida útil de Angra 1 e retomada de Angra 3.” .

- [11] DREISEITL, S., OHNO-MACHADO, L., 2002, “Logistic regression and artificial neural network classification models: a methodology review”, *Journal of Biomedical Informatics*, v. 35, n. 5, pp. 352–359. ISSN: 1532-0464. doi: [https://doi.org/10.1016/S1532-0464\(03\)00034-0](https://doi.org/10.1016/S1532-0464(03)00034-0). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1532046403000340>>.
- [12] DSA, D. S. A., 2018, *Deep Learning Book*. <https://www.deeplearningbook.com.br/>.
- [13] EIA, U. E. I. A., 2014, “www.eia.gov/todayinenergy/detail.php?id=19091”, .
- [14] ELLER, I. B., 2018, “Uma Metodologia para Avaliação Regulatória de Extensão de Vida de Usinas Nucleares.” .
- [15] GOODFELLOW, I.; BENGIO, Y. C. A., 2016, *Deep Learning*. MIT Press.
- [16] HAN, J., MORAGA, C., 1995, “The influence of the sigmoid function parameters on the speed of backpropagation learning”. In: Mira, J., Sandoval, F. (Eds.), *From Natural to Artificial Neural Computation*, pp. 195–201, Berlin, Heidelberg. Springer Berlin Heidelberg. ISBN: 978-3-540-49288-7.
- [17] HEBB, D. O., 1949, *The Organization of Behavior*. John Wiley And Sons, Inc.
- [18] HOCHREITER, S., SCHMIDHUBER, J., 1997, “Long Short-Term Memory”, *Neural Computation*, v. 9, n. 8, pp. 1735–1780. doi: 10.1162/neco.1997.9.8.1735.
- [19] HORNIK, , STINCHOMBE, M., 1992, “Multilayer feed-forward networks are universal approximators”, .
- [20] HU, Y., HUBER, A., ANUMULA, J., et al., 2019. “Overcoming the vanishing gradient problem in plain recurrent networks”. .
- [21] IAEA, 2010, ““Periodic Safety Review of Nuclear Power Plants”. Revision of NS-G-2.10 Safety Guide.” .
- [22] IAEA, I. A. E. A., 1998, “EQUIPMENT QUALIFICATION IN OPERATIONAL NUCLEAR POWER PLANTS: UPGRADING, PRESERVING AND REVIEWING”, .
- [23] IAEA, I. A. E. A., 2009, “Periodic Safety Review of Nuclear Power Plant”, .

- [24] JAIN, A. K., M. J. M. K., 1996, “Artificial neural networks: a tutorial”, *IEEE Computer*, pp. 56–63.
- [25] LOPES, N. S., 2011, “Avaliaç~ao do uso de Redes Neurais e Regress~ao Linear M´ultipla na recomposiç~ao de dados atmosf´ericos de estaç~oes costeiras da Marinha do Brasil”, .
- [26] LUKOŠEVIČIUS, M., JAEGER, H., 2009, “Reservoir computing approaches to recurrent neural network training”, *Computer Science Review*, v. 3, n. 3, pp. 127–149. ISSN: 1574-0137. doi: <https://doi.org/10.1016/j.cosrev.2009.03.005>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1574013709000173>>.
- [27] MARSAGLIA, G., 2004, “Evaluating the Normal Distribution”, .
- [28] MCCULLOCH, W. S., PITTS, W., 1943, “A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics”, .
- [29] MICHALSKI, R. S., TECUCI, G., 1994, “Machine Learning: A Multistrategy Approach, Volume IV”, .
- [30] MILAN, N. S. P., 1987, “Nuclear power plant ageing and life extension: Safety aspects An overview of issues and the IAEA’s symposium”, .
- [31] MINSKY, M., PAPER, S., 1969, *Perceptrons. An Introduction to Computational Geometry*. M.I.T. Press, Cambridge, Mass.
- [32] NRC, N. R. C., 1954, “REQUIREMENTS FOR RENEWAL OF OPERATING LICENSES FOR NUCLEAR POWER PLANTS”, .
- [33] OLAH, C., 2015, “Understanding LSTM Networks”, Disponível em: <<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>>.
- [34] PINHEIRO, V. H. C., 2020, “Redes Neurais de Aprendizado Profundo Aplicadas a Problemas Complexos da Engenharia Nuclear”, .
- [35] ROSENBLATT, F., 1957, “The perceptron, a perceiving and recognizing automaton”, .
- [36] RÓNAKY, J., 2007, “Environmental qualification of equipment during the design of nuclear power plants”, .
- [37] SCHIRRU, P., 2014, “A Real Time Artificial Intelligent Monitoring System for Nuclear Power Plants Operators Support”, pp. 71–83.

- [38] SHAH, D., CAMPBELL, W., ZULKERNINE, F. H., 2018, “A Comparative Study of LSTM and DNN for Stock Market Forecasting”, *2018 IEEE International Conference on Big Data (Big Data)*, pp. 4148–4155.
- [39] SHEWALKAR, A. N., 2018, *Comparison of RNN, LSTM and GRU on Speech Recognition Data*. M.Sc. dissertation, North Dakota State University of Agriculture and Applied Science.
- [40] SPEARMAN, C., 1904, “The Proof and Measurement of Association between Two Things”, .
- [41] SPECHT, D. F., 1991, “A General Regression Neural Network”, pp. 568–576.
- [42] SQUARTINI, S., HUSSAIN, A., PIAZZA, F., 2003, “Preprocessing based solution for the vanishing gradient problem in recurrent neural networks”. In: *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03.*, v. 5, pp. V–V. doi: 10.1109/ISCAS.2003.1206412.
- [43] TODREAS, N., 2015, “How nuclear power-generating reactors have evolved since their birth in the 1950s”, .
- [44] WERBOS, P. J., 1990, “Backpropagation through time: what it does and how to do it”, *IEEE*, pp. 1550–1560.
- [45] WNA, W. N. A., 2019, “Methodology to Manage Material and Waste from Nuclear Decommissioning”, .