RECONSTRUÇÃO DA DISTRIBUIÇÃO DE DENSIDADE DE POTÊNCIA RADIAL EM PWRs A PARTIR DAS LEITURAS DO SDNI USANDO REDES NEURAIS ARTIFICIAIS.

Álefe Calebe Ribeiro Lima

Dissertação 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.

Orientadores: Prof. Alessandro da Cruz Gonçalves
Prof. Aquilino Senra Martinez

Rio de Janeiro
Fevereiro de 2023

RECONSTRUÇÃO DA DISTRIBUIÇÃO DE DENSIDADE DE POTÊNCIA RADIAL EM PWRs A PARTIR DAS LEITURAS DO SDNI USANDO REDES NEURAIS ARTIFICIAIS.

Álefe Calebe Ribeiro Lima

DISSERTAÇÃO APRESENTADA 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 DA ENGENHARIA NUCLEAR.

Orientadores: Prof. Alessandro da Cruz Gonçalves
Prof. Aquilino Senra Martinez

Examinada por: Prof. Alessandro da Cruz Gonçalves
Prof. Aquilino Senra Martinez
Dr. Edson Henrice Júnior
Prof. Zelmo Rodrigues de Lima

RIO DE JANEIRO, RJ - BRASIL
FEVEREIRO DE 2023

Lima, Álefe Calebe Ribeiro

Reconstrução da Distribuição de Densidade de Potência Radial em PWRs a Partir de Leituras do SDNI Usando Redes Neurais /Alefe Calebe Ribeiro Lima. – Rio de Janeiro: UFRJ / COPPE, 2023

XVI, 129 p.: il..; 29,7 cm.

Orientadores: Alessandro da Cruz Gonçalves.

Aquilino Senra Martinez.

Dissertação – UFRJ/ COPPE/ Programa de Engenharia Nuclear, 2023.

Referências Bibliográficas: p. 101-109.

1. xxxxxxxxxxxxxxxxxx. 2. xxxxxxxxxxxxxxxxxx. 3. xxxxxxxxxxxxxxxxxx. I. Gonçalves, Alessandro da Cruz *et al*. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Nuclear. III. Título.

Agradecimentos

Agradeço a minha família, Marcia, Nivaldo e Marcelle, em especial a meus pais, por terem me apoiado com muita luta, esforço e dedicação no decorrer da minha vida, principalmente na escolha de ingressar no mestrado, sendo os grandes responsáveis por eu ter chegado até aqui.

Agradeço a minha namorada e companheira Karoline, que me apoio e ajudou em todos momentos no decorrer de 5 anos.

Agradeço ao professor Alessandro da Cruz Gonçalves, por me despertar o interesse pela área, pela orientação e dedicação que sem elas o trabalho não teria sido possível.

Agradeço ao professor Aquilino Senra, por toda ajuda prestada e pelas ideias e sugestões no decorrer do trabalho, ao professor Adilson, pelos conselhos e ensinamentos transmitidos.

Agradeço ao Doutor Edson Henrice, por todos ensinamentos no código SERPENT, ensinamentos em tópicos da física de reatores, como também auxilio nas implementações das redes neurais artificiais que possibilitaram a realização deste trabalho.

Agradeço aos meus amigos de pesquisa do código SERPENT João e Yan que estiveram comigo no decorrer deste trabalho.

Gostaria de agradecer a todos funcionários da secretaria do Programa de Engenharia Nuclear, em especial Cassia e Reginaldo, que estiveram sempre a disposição para ajudar em qualquer necessidade.

Por fim agradeço a todos que de forma direta ou indiretamente contribuíram para a realização deste trabalho.

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

RECONSTRUÇÃO DA DISTRIBUIÇÃO DE DENSIDADE DE POTÊNCIA RADIAL EM PWRs A PARTIR DAS LEITURAS DO SDNI USANDO REDES NEURAIS ARTIFICIAIS.

Álefe Calebe Ribeiro Lima

Fevereiro/2023

Orientadores: Alessandro da Cruz Gonçalves
Aquilino Senra Martinez

Programa: Engenharia Nuclear

Os reatores do tipo PWR que utilizam esquemas de carregamento nuclear para minimizem a fuga de nêutrons são altamente heterogêneos e tem margens operacionais reduzidas. Portanto, é fundamental a previsão precisa da distribuição de densidade de potência, a fim de aumentar as margens de manobras operacionais.

A presente dissertação visa o desenvolvimento de uma metodologia para reconstrução da distribuição de potência radial a partir de técnicas de redes neurais artificias utilizando das medidas fornecidas pelo Sistema de Detecção Nuclear Interna de um reator nuclear similar ao de Angra-1, simulados através do código de transporte Monte Carlo.

A metodologia apresenta-se viável, considerando a proximidade de valores da reconstrução da distribuição de potência com os valores simulados. Consequentemente contribuindo para a expansão da margem de manobras operacionais de reatores do tipo PWR.

.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

RECONSTRUCTION OF THE RADIAL POWER DENSITY DISTRIBUTION IN PWRs FROM SDNI READINGS USING ARTIFICIAL NEURAL NETWORKS

PWR reactors that uses nuclear charging schemes to minimize the neutrons scape are highly heterogeneous and has operational margins reduced. Therefore, is fundamental the accurate prediction of the power density distribution, in order to increase the operational margins for maneuvers.

This thesis aims the development of a methodology for the reconstruction of radial power from artificial neural networks technics using the statistics given by the Internal Nuclear Detection System from a nuclear reactor similar to the Angra-1 reactor, simulated by the Monte Carlo transport code.

The methodology presents viable considering the proximity of the power distribution reconstruction values with the simulated values. Pursuant to, contributing to the expansion of the operational margin of maneuvers of PWR reactors.

Álefe Calebe Ribeiro Lima

February/2023

Advisors: Alessandro da Cruz Gonçalves
Aquilino Senra Martinez

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

MPL	*Multilayer Perceptron*
PWR	*Pressurized Water Reactor*
RNA	Rede Neural Artificial
SDNI	Sistema de Detecção Nuclear Interna
SGD	*Stochastic Gradient Descent*
SMR	*Small Modular Reactor*

1. Introdução

Nos últimos anos é perceptível o aumento do consumo de energia por diversos motivos, sendo o aumento exponencial populacional um dos mais notáveis (Oliveira, 2016). Em função disso, Oliveira (2016) expõe a necessidade da ampliação da matriz energética tendo em consideração a sustentabilidade, uma vez que se torna mais necessário no atual contexto mundial a da redução de emissões do dióxido de carbono (CO_2).

Em sua exposição, o autor menciona que a fim de suprir a demanda energética que cresce constantemente com menor impacto ambiental possível, são considerados métodos de geração de energia, sendo um deles, o de geração através da energia nuclear. Nesse contexto, a energia nuclear se destaca como uma das possíveis alternativas, pois trata-se também de uma fonte energia limpa de baixa emissão de CO2, com a vantagem de ser uma energia segura e de alto fator de capacidade. Além disso, tendo em vista o avanço das tecnologias dos reatores nucleares de quarta geração e dos SMRs (*Small Modular Reactor*), faz com que a energia nuclear se torne um possível pivô para a redução de CO2, tornando-se uma proeminente fonte energia sustentável.

À vista disto, Duderstadt (1976) registra que não somente o fator de baixo impacto ambiental coopera para o emprego da energia nuclear na sociedade, como também apresenta uma grande reserva de combustíveis, além de possuir vantagens operacionais e financeiras em comparação com as fontes de energia convencionais.

Portanto, é de suma importância o contínuo aprimoramento das tecnologias dos reatores nucleares, a fim de garantir uma operação cada vez mais segura, mitigando acidentes (Souza *et al*, 2017), e, assim tornando a energia nuclear mais competitiva não somente no que diz respeito a sustentabilidade como também economicamente.

No contexto das tecnologias de reatores nucleares de segunda e terceira geração os Reatores a Água Leve Pressurizada - PWR foram os que se mostraram mais promissores, constituindo a maior parte dos reatores nucleares em atividade no mundo de acordo com IAEA (2021). Portanto, o desenvolvimento de tecnologia e metodologias que visam a contínua monitoração de parâmetros do reator torna-se fundamental para garantir máxima segurança operacional desses reatores. Por exemplo, em reatores PWR a monitoração da distribuição da densidade de potência durante a operação é importante para a determinação de condições seguras de operação, permitindo uma maior margem

para manobras operacionais (Makai e Végh, 2017), sendo altamente relevante uma vez que possibilita que a queima possa atingir níveis elevados.

Assim, a monitoração do reator, com o objetivo de obter informações precisas e detalhadas da condição do núcleo do reator, é de extrema importância, garantindo melhor manuseio do combustível e também uma operação flexível e eficiente (Saeed e Rashid, 2020).

A esse respeito, Li *et al* (2017) aponta que monitoração em tempo real da distribuição de densidade de potência é uma das metodologias mais importantes para um reator nuclear, dado que a distribuição de potência é um parâmetro crucial para operação segura do reator. Além disso, a monitoração em tempo real da distribuição de potência possibilita uma melhor gama de manobras operacionais do reator.

Entretanto, segundo Souza e Moreira (2006) na maioria dos reatores do tipo PWR a monitoração da distribuição de potência do núcleo do reator é realizada através do Sistema de Detecção Nuclear Externo (ex-*core*). Tendo isso em vista, os autores que os recursos para estimar a distribuição de potência em tempo real a partir do Sistema de Detecção Nuclear Externo, possuem incertezas de até 20%, em consequência, torna mais restritiva a margem de operação destes reatores.

Contudo, a realização da monitoração em tempo real da densidade de potência através do Sistema de Detecção Nuclear Interna (SDNI), é possível alcançar uma melhor margem operacional de manobras do reator.

Portanto, a realização da monitoração em tempo real através implantação destes detectores nos reatores mais modernos levou ao desenvolvimento de diversos sistemas de monitoração nuclear, a fim de determinar a distribuição de potência a partir de medidas fornecidas pelo SDNI (Rashid e Rashid, 2020).

Assim, tendo em vista tais tecnologias, a reconstrução da distribuição de densidade de potência em todos os elementos combustíveis a partir das medidas fornecidas pelo SDNI, torna possível a estimativa de medidas representativas das condições físicas e termohidráulicas do reator, e, com isso ampliando as margens operacionais do reator.

Na literatura científica é possível encontrar ampla diversidade de técnicas para monitoração de distribuição de potência (Makai e Végh, 2017) nas quais são baseadas na previsão da distribuição da densidade de potência a partir dos sinais fornecidos pelo SDNI, os quais são inseridos em pontos estratégicos do núcleo do reator.

Nesse conceito, um exemplo de tais técnicas é o método adaptativo, que corrige a distribuição da densidade de potência fornecida por um simulador neutrônico e por meio dos sinais dos detectores internos, como no caso do sistema BEACON (Boyd and Miller, 1967) que segundo Saeed e Rashid (2020) foi desenvolvido pela Westinghouse para reatores do tipo PWR. O BEACON utiliza dos valores fornecidos pelo SDNI em conjunto com um método nodal 3D analítico, para medição e análise da distribuição de potência em tempo real.

Ademais, outros métodos podem ser citados, como por exemplo o método do mapeamento racional das medições dos detectores internos (Bonalumi *and* Kherani, 1985), que consiste basicamente na utilização de um modelo iterativo que envolve a equação da difusão de nêutrons e sinais dos detectores; o método dos resíduos ponderados, que busca uma estimativa da densidade de potência a partir de uma técnica de ponderação de resíduos, em que os resíduos são as diferenças entre o valor real e valor obtido pela expansão em autofunções, tal método busca definir um conjunto de funções de ponderação que minimizem os resíduos obtendo-se, finalmente, os coeficientes da expansão em autofunções que estimam a distribuição da densidade de potência.

Dessa forma, esse trabalho apresenta-se relevante, uma vez que a implantação de um sistema computacional para acompanhamento da densidade de potência em tempo real, aumenta as margens de segurança operacional do reator, simplificando a necessidade de um sistema de detecção em todos os elementos combustíveis, diminuído assim o custo com a instrumentação nuclear.

Ainda nesse contexto, o trabalho visa o desenvolvimento de uma metodologia baseada em técnicas de redes neurais artificiais para reconstrução da distribuição de densidade de potência em reatores do tipo PWR, a partir das medidas fornecidas pelo SDNI, obtidas através de simulações computacionais. Desta maneira, visando expandir a margem para manobras de operação de reatores nucleares similares ao de Angra-1.

Para alcançar seus objetivos, o trabalho será dividido em seis seções, incluindo a introdução e a conclusão, para além dos apêndices e referências. Logo, sendo o capítulo 1 introdutório, o capítulo 2 introduz o código SERPENT, utilizado para simular um reator PWR similar à Angra-1, a fim de simular as respostas de densidade de potência adquiridas por meio do Sistema de Detecção Nuclear Interno, as quais são utilizadas para reconstrução da distribuição de potência do núcleo do reator, por meio da utilização das técnicas de redes neurais.

No capítulo 3 é apresentada uma breve descrição dos principais métodos de redes neurais artificiais, em especial o método empregado nesta pesquisa, com o objetivo de reconstruir a distribuição da densidade de potência do núcleo do reator.

No capítulo 4 é descrito detalhadamente a aplicação da técnica de redes neurais artificiais de múltiplas camadas, para a reconstrução da distribuição de potência a partir das respostas do SDNI, simuladas pelo código SERPENT.

No capítulo 5 são discutidos os resultados alcançados pela metodologia proposta. E, por fim, o último capítulo, capítulo 6, as conclusões e perspectivas futuras acerca do trabalho são registradas.

2. Simulação do reator de referência com o código SERPENT

O método de Monte Carlo fornece valores médios de determinadas grandezas físicas obtidos por meio de simulações contabilizando dados coletados a partir da simulação do caminho percorrido por partículas através de um meio.

Diante disso, Yoriyaz (2009) indica que o método de Monte Carlo se configura em um cálculo estatístico onde emprega uma sequência de números aleatórios para a realização da simulação, onde muitas vezes o processo físico é simulado diretamente sem a necessidade das equações que o descrevem. Além disso, o autor expõe que em casos de transporte de partículas, o conjunto de eventos sofridos pela partícula, desde a sua emissão até o momento que é absorvida, é intitulado de história da partícula, e que tais histórias são geradas por meio de funções densidade de probabilidade. Neste sentido, em aplicações em problemas na física de reatores, o processo é similar.

Segundo Leppänen (2007), são simuladas as interações físicas dos nêutrons com a matéria (nuclídeos) e seu caminho percorrido a partir da sua emissão até sua absorção ou fuga do sistema. Sobre isso, Mickus (2021) explica que as histórias são simuladas, ao realizar o rastreamento de cada nêutron durante uma série de eventos probabilísticos que descrevem a interação dos nêutrons com a matéria. Após uma grande quantidade de histórias de nêutrons simuladas, são deduzidas as características do sistema simulado.

Assim, uma vez que as interações dependem da energia inicial do nêutron, temperatura do ambiente e o tipo de material envolvido nas colisões, são empregados dados fornecidos por bibliotecas de dados nucleares, como a *Evaluated Nuclear Data File* (ENDF), e, por fim sendo calculado todo o processo de transporte dos nêutrons (Leppänen, 2007).

Para o desenvolvimento deste projeto de pesquisa foi escolhido o Código de Monte Carlo de Física de Reatores SERPENT (Leppänen *et al*. 2015), desenvolvido no VTT *Technical Research Center of Finland (Valtion Teknillinen Tutkimuskeskus)*. O SERPENT (Leppänen *et al*. 2015), originalmente foi desenvolvido como um código para Física de Reatores, possibilitando práticas como: modelagem de reator, geração de constantes de grupo como seção de choque macroscópica homogeneizada, coeficientes de difusão, queima combustível etc.

O código SERPENT foi escolhido dentre os demais códigos de Monte Carlo pois além do Laboratório de Engenharia Nuclear possuir uma licença deste código, possui expertise na sua utilização (Henrice *et al*., 2017; Henrice Jr. *et al*., 2021; Oliveira *et al*., 2019; Ribeiro *et al*., 2020). Este código foi criado especificamente para física de reatores com funcionalidades que facilitam tanto a modelagem computacional das estruturas materiais que compõe o núcleo do reator, quanto facilitam a obtenção dos parâmetros macroscópicos necessários para inúmeras aplicações, como por exemplo a resolução da equação difusão de nêutrons.

Outra vantagem significativa do código SERPENT é a utilização do cálculo paralelo. De acordo com Leppänen (2007), a linearidade do processo de transporte de nêutrons, que tem como característica o comportamento dos nêutrons que mantem a reação de fissão em cadeia, interagirem apenas com o meio e não entre eles. Tornando assim, possível a utilização do cálculo paralelo na simulação do código.

A respeito disso, o autor expressa que o cálculo paralelo tem grande importância na computação cientifica, sendo uma opção viável na realização de tarefas com grande custo computacional, permitindo o tratamento de problemas, de difícil abordagem. A implementação da paralelização no código SERPENT, é feita através do Message Passing Interface (MPI), que é uma biblioteca que apresenta alto grau de cálculo paralelo, comunicação e compartilhamento de dados entre as tarefas. Dessa forma possibilitando a realização mais rápida dos cálculos.

Portanto, o tempo geral de cálculo, comparado a outros programas que utilizam o Método de Montecarlo, utilizando a mesma estrutura de hardware, é reduzido significativamente, em alguns casos reduzido próximo a duas ordens de grandeza (Leppänen, 2009).

Quanto a modelagem computacional, o SERPENT possui funcionalidades que facilitam a simulação das geometrias que compõem o reator e suas composições, como inserir o tamanho do pitch das varetas no elemento combustível, como também definição de universos de pino que de acordo com o código são regiões anulares aninhadas, tais universos geralmente são usados na construção de varetas combustíveis, mas podem também ser usados para construção de varetas de veneno queimável e barras de controle, etc. A figura 2.1 apresenta a composição da vareta combustível a ser simulada no código.

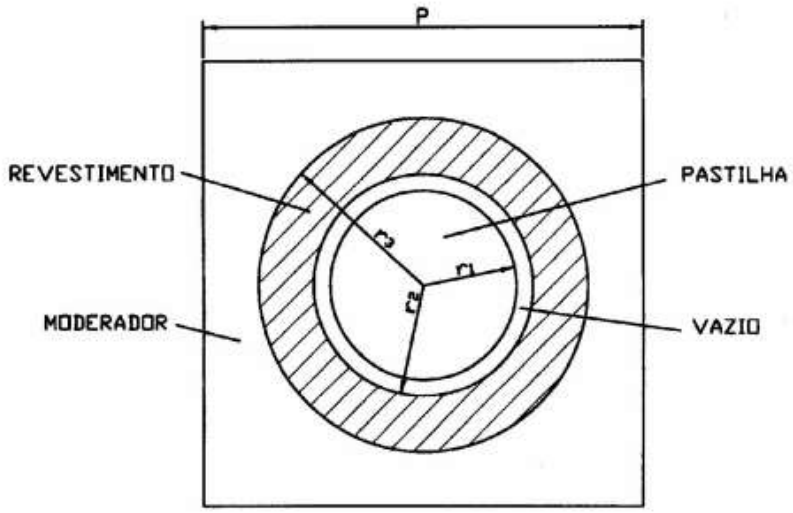

Figura 2.1: Composição da vareta combustível.

Fonte: Sadde (2000).

Dessa forma, na construção dos universos de pino, e necessário a inserção dos materiais que o compõem e seus respectivos raios, na vareta combustível exposta na Figura 2.1, é necessário declarar três tipos de materiais e seus raios.

Todos os materiais a serem simulados, necessitam ser definidos no código, o SERPENT necessita de parâmetros para a declaração de um material, sendo indispensável a inserção das densidades do material e dos nuclídeos que o formam, a tabela 2.1 apresenta a composição de uma pastilha combustível.

Tabela 2.1: Composição material de uma pastilha combustível.

Fonte: Elaboração própria.

Composição Pastilha combustivel	
Densidade pastilha	10,4091886 g/cm³
Nuclídeos	Densidade atômica
U-234	3,54100E+18
U-235	4,9367E+20
U-236	3,04E+18
U-238	2,2717E+22
O-16	4,6434E+22

Assim, com as declarações dos materiais é possível a construção das varetas combustíveis e demais componentes do sistema a ser simulado, após a construção dos

universos de pino, é realizável a construção dos elementos combustíveis, que são construídos utilizando o conceito de *lattices*, que é uma palavra inglesa que pode expressar um arranjo de estruturas repetidas, na física de reatores Leppänen (2015) explicita que a palavra *lattice* define a estrutura repetitiva dos materiais físseis e não-físseis.

A *lattice* apresentada pelo código SERPENT (Leppänen *et al*. 2015) é definido como um tipo especial de universo preenchido com universos repetidos, dessa forma na construção de um elemento combustível este conceito é aplicado, onde uma estrutura é construída com os universos dos pinos que formam as varetas combustíveis, ou seja, as varetas combustíveis de um determinado EC, são definidas separadamente e após inseridas em suas referentes posições no elemento combustível. A Figura 2.2 apresenta um exemplo de *lattice* do código.

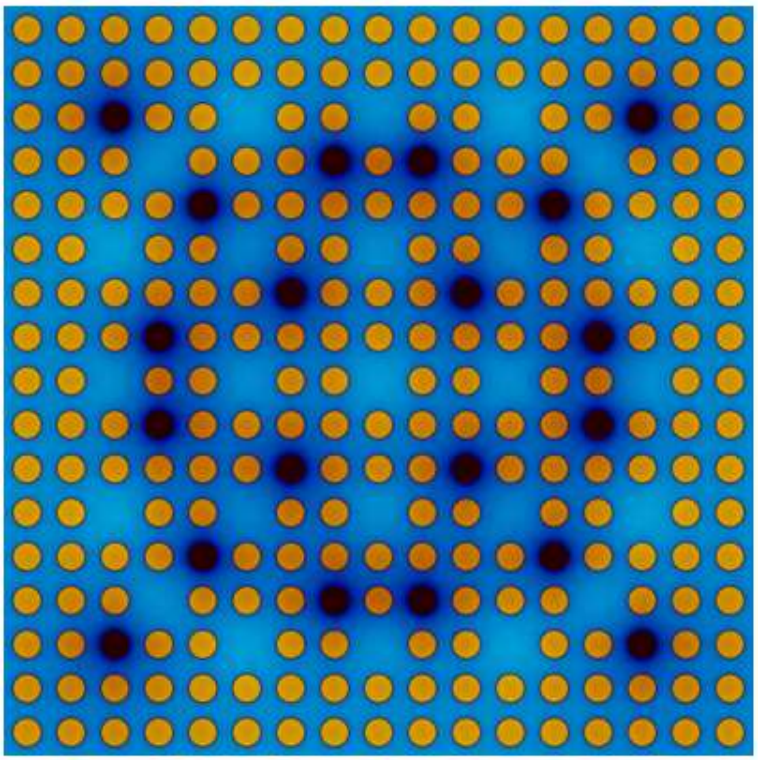

Figura 2.2: Distribuição dos pinos em uma *lattice*.

Fonte: Leppänen *et al*. (2015).

Assim para a construção do núcleo do reator é criada uma *lattice* que é preenchida com as *lattices* dos elementos combustíveis.

Dessa forma, o código SERPENT consegue modelar e simular um reator do tipo PWR, utilizado nesta pesquisa para simular um reator similar a Angra-1, considerando três zonas de enriquecimento, conforme mostrado na Figura 2.3.

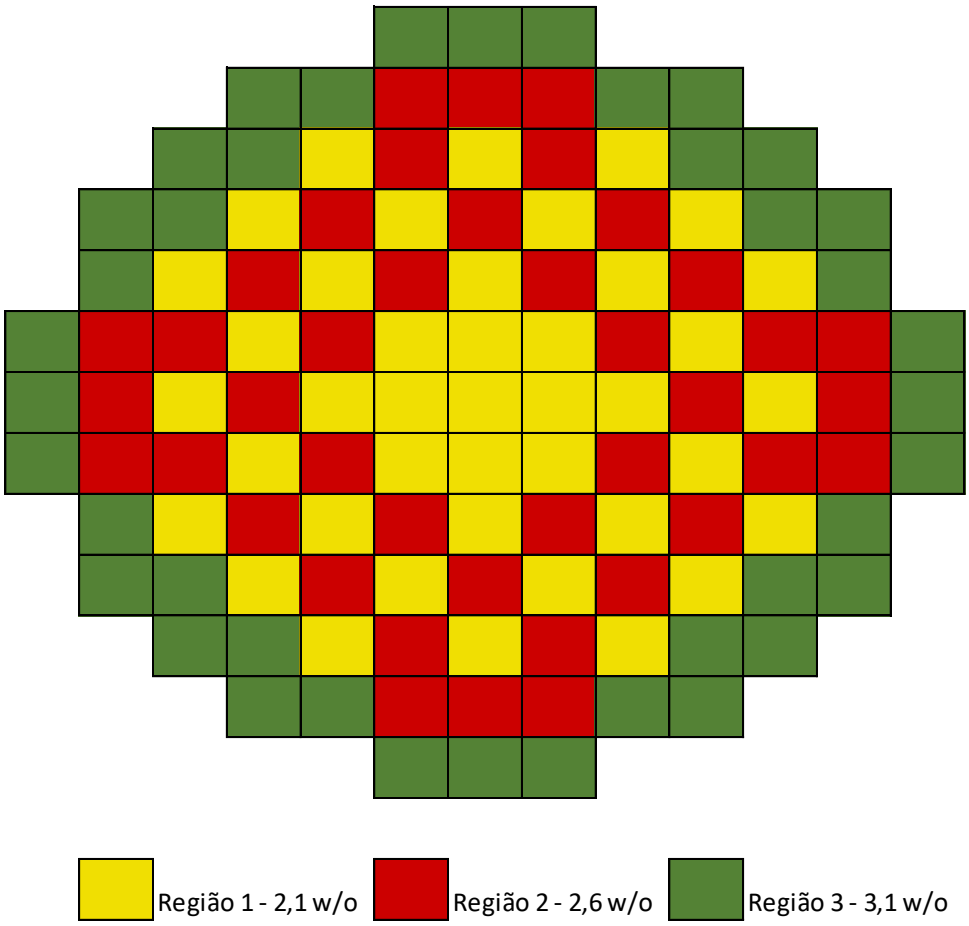

Figura 2.3: Núcleo de um reator similar a Angra-1 com três zonas de enriquecimento.
Fonte: Elaboração própria.

Para tal, o código SERPENT é utilizado para simular as densidades de potência nos elementos combustíveis que contém os 36 detectores do SDNI similar ao do reator de Angra-1, vide Figura 2.4

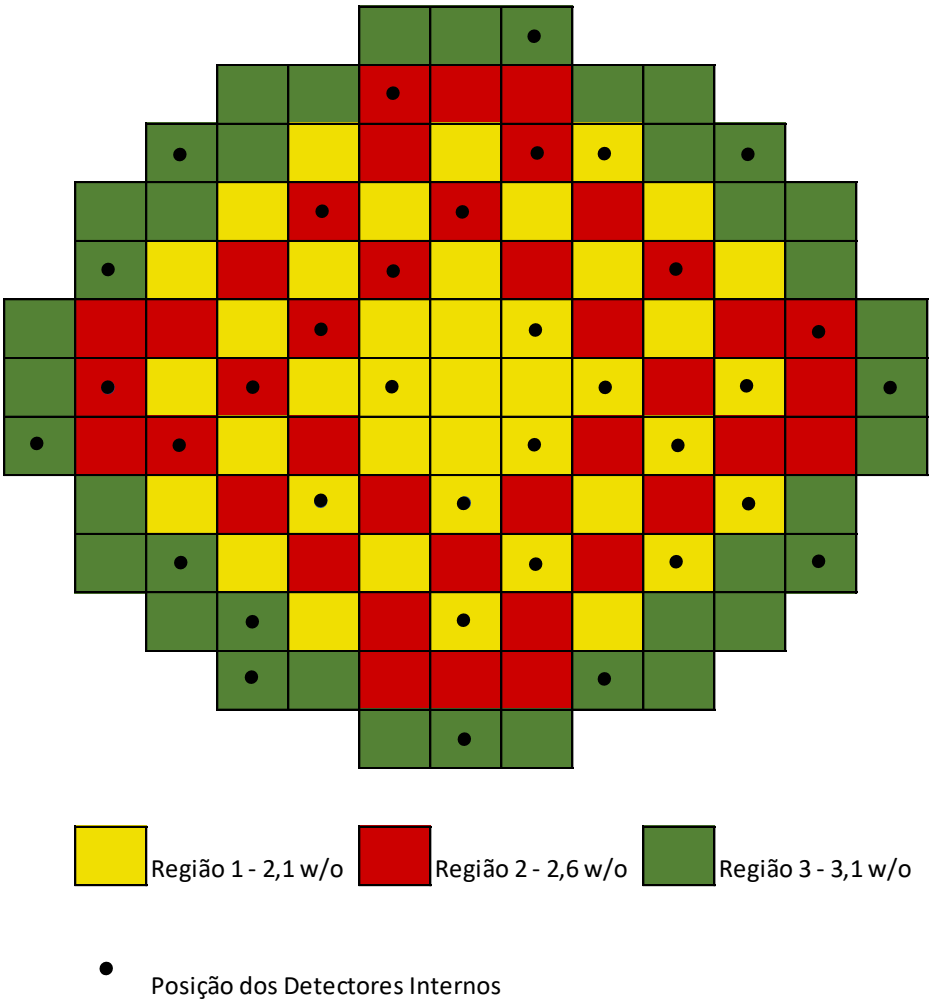

Figura 2.4: Posição dos detectores internos no núcleo do reator similar a Angra-1

Fonte: Elaboração própria.

O código de simulação SERPENT (Leppänen *et al*. 2015) permite implementar detectores no sistema de diversas formas, podendo ser ele implementado nas *lattices*, como elementos combustíveis ou núcleo, ou por divisão em forma de malha. Além disso, é oferecido pelo código a obtenção de diversos parâmetros nucleares, como taxa de fissão, fluxo, taxa de fuga, taxa de captura, distribuição de densidade de potência, etc.

Com isso em vista, a simulação é configurada para fornecer valores de potência em todos os elementos combustíveis do núcleo do reator.

A fim de se obter máxima precisão nos valores simulados, o sistema é simulado juntamente com a queima dos elementos combustíveis. A tabela 2.2 apresenta os dias de queima e a queima combustível acumulada.

Tabela 2.2: Instantes de queima e queima combustível acumulada.

Fonte: Elaboração própria.

Dias de operação em plena potência x Queima	
Dias	acumulada (MWD/MTU)
0	0
3,99	150,65
13,33	503,30
26,65	1006,20
39,98	1509,50
53,3	2012,40
79,96	3019,10
106,61	4025,30
133,26	5031,50
159,91	6037,70
186,57	7044,30
213,22	8050,50
239,87	9056,80
266,52	10063,00
293,18	11070,00
319,83	12076,00
333,16	12579,00
346,48	13082,00

Com isso em vista, a simulação do núcleo do reator é viável, uma vez que a metodologia do SERPENT (Leppänen *et al.*, 2015), disponibiliza exemplos de queima de um elemento combustível, assim torna possível a realização de um processo semelhante para a realização da queima do núcleo inteiro.

Para isso, foi necessário buscar metodologias para obter valores dos parâmetros que ao serem inseridos no código para simulação, permitam ao código encontrar dados de operação aceitáveis para uma operação segura e contínua do reator, tais como o fator de multiplicação de nêutrons, concentração de boro, queima combustível, etc. Como também, para encontrar similaridade dos valores de densidade de potência obtidos pela simulação, com os valores de densidade de potência presentes no manual de operação do ciclo 1 de Angra-1.

Dessa forma, no processo de estruturar o código para realizar a simulação de todo o núcleo do reator, fez-se necessário a realização de diversas simulações de teste, com o objetivo de otimizar a codificação da simulação e servir de referência para futuras modificações.

Logo, foram necessárias simulações para comparação e ajuste de comandos referentes a queima, fluxo de nêutrons, densidade de potência entre outros. Sendo necessária uma considerável quantidade de tempo para cada simulação teste, já que para o cálculo da queima, Leppänen (2007) esclarece que é necessário um imenso esforço computacionalmente, porque a equação de transporte é calculada para uma grande quantidade de passos de depleção.

Assim, sendo uma simulação realizada, os parâmetros por ela obtidos, foram comparados com valores dos parâmetros presentes no manual de operação do primeiro ciclo de Angra-1.

Desta forma, para realizar a simulação é preparado o modelo de um reator similar a Angra-1, no qual é a fim de obter a distribuição de densidade de potência em diversos instantes de queima de operação, para obter os valores de potência de todo núcleo do reator como os valores medidos pelo do Sistema de Detectores Nucleares Internos.

Na preparação do modelo a ser simulado é definida a composição dos materiais do sistema e seus arranjos geométricos como exposto na Tabela 2.1 onde são levados em consideração a temperatura e pressão do material para a declaração de sua densidade. Como previamente mencionado, o código SERPENT, destaca-se pela facilidade na construção de estruturas repetidas para simulação. Em vista disso, Leppänen (2009), elucida que a implantação da geometria no SERPENT, consiste de células materiais homogêneas, definidas por uma combinação de superfícies elementares que combinadas formam as geometrias do reator. O autor descreve que utilizando apenas superfícies planas e cilíndricas é possível definir qualquer geometria do reator. Leppänen (2009) esclarece que diversos códigos dispõem da implementação de superfícies como primas e quadrados, que apesar de facilitar o usuário no momento da declaração das geometrias, pode levar a problemas de eficiência consideráveis.

Entretanto, para definição das estruturas repetidas como varetas combustíveis e elementos combustíveis, o autor aponta que a declaração de cada unidade geométrica independente, não é a maneira mais eficiente. Dessa forma, esses tipos de geometrias são construídos utilizando o conceito de universos e *lattices*. Assim a definição dos elementos combustíveis e da configuração de carregamento do núcleo, possuem menor tempo de declaração no código.

Portanto, na definição da geometria do reator similar ao reator de referência Angra-1, são definidas as geometrias do vaso do reator, proteção térmica, refletores e disposição do núcleo, que conta com elementos combustíveis de três diferentes níveis de

enriquecimento de Urânio-235, sendo 41 elementos combustíveis com 2,1% de enriquecimento, 40 com 2,6% de enriquecimento 40 elementos com 3,1% de enriquecimento sendo um ciclo fresco, sem elemento combustível queimado. Ainda na construção do modelo, considera-se as disposições das barras de veneno queimável e suas quantidades constando elementos combustíveis com 8, 12 e 16 varetas de veneno queimável.

A Figura 2.5 apresenta ¼ do núcleo simulado, modelado pelo código SERPENT.

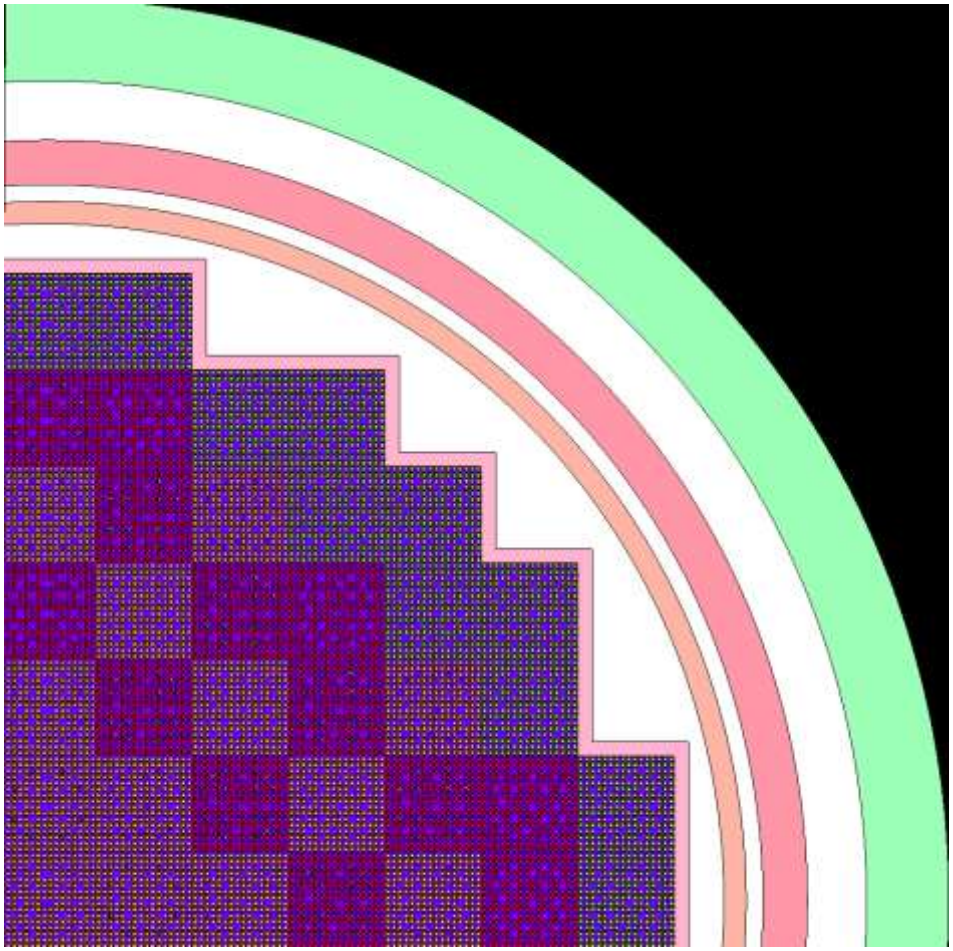

Figura 2.5: ¼ do núcleo modelado pelo código SERPENT

Fonte: Elaboração própria.

Além disso, foi determinado os elementos combustíveis que compõem os bancos de barra de controle, com tubos guias para a inserção das barras. Mas a simulação foi realizada sem nenhuma inserção de banco de barras de controle.

A fim de encontrar as densidades necessárias, foram usadas métodos presentes em literaturas como Freitas (2017), que apresenta um modelo de cálculo para obtenção da

densidade teórica de uma pastilha combustível, Lamarsh (1983), que expõe a metodologia permite encontrar as densidades dos isótopos que compõem a pastilha combustível de $U0_2$ para os referentes enriquecimentos e também Godfrey (2014), no qual disponibiliza o método de calcular as densidades do boro a partir das frações de peso do pyrex de borossilicato.

Por fim, após as inserções necessárias para a criação do modelo, realizou-se a simulação do reator, que apresentou resultados satisfatórios quando comparadas com dados presentes no manual de operação de Angra-1, em que os valores comparados foram a distribuição de densidade de potência, queima de materiais combustíveis e concentração de boro no final da operação. Foram simulados 18 instantes de queima, referentes aos intervalos dos dias efetivos de plena potência 3,99; 13,33; 26,65; 39,98; 53,30; 79,96; 106,61; 133,26; 159,91; 186,57; 213,22; 239,87; 266,52; 293,18; 319,83; 333,16 e 346,48.

As figuras 2.6, 2.7, 2.8 e 2.9 apresentam comparação dos valores de potência normalizados de 1/8 do núcleo presentes no manual de operação de Angra-1 com os valores simulados pelo código SERPENT (em amarelo) e seus erros relativos percentuais, referentes aos intervalos de queima 53,30 dias, 106,61 dias, 159,91 dias e 213,22 dias.

O erro relativo percentual é calculado da seguinte forma:

$$E = \frac{|V_M - V_S|}{V_M} \text{x} 100 \qquad (2.1).$$

Onde:

E: Erro relativo percentual;

V_M: Valor presente no manual de operação;

V_S: Valor simulado pelo código SERPENT.

1,21				
1,10				
9%				
1,13	1,19			
1,08	1,11			
4%	7%			
1,18	1,14	1,18		
1,12	1,18	1,15		
5%	4%	3%		
1,13	1,16	1,16	1,09	
1,18	1,14	1,18	1,09	
4%	2%	2%	0%	
1,13	1,12	1,07	1,01	0,66
1,12	1,16	1,05	1,04	0,69
1%	4%	2%	3%	5%
1,03	1,04	0,9	0,6	
1,06	0,99	0,86	0,61	
3%	5%	4%	2%	
0,71	0,58			
0,71	0,59			
0%	2%			

Figura 2.6: Comparação dos valores e seus respectivos erros relativos percentuais para o intervalo de queima de 53,30 dias efetivos de plena potência.

Fonte: Elaboração própria.

1,21				
1,12				
7%				
1,16	1,2			
1,10	1,14			
5%	5%			
1,19	1,18	1,18		
1,14	1,21	1,17		
4%	3%	1%		
1,16	1,17	1,17	1,08	
1,22	1,15	1,19	1,06	
5%	2%	2%	2%	
1,13	1,13	1,05	0,99	0,65
1,11	1,15	1,02	1	0,64
2%	2%	3%	1%	2%
1,03	1,02	0,88	0,59	
1,03	0,95	0,81	0,56	
0%	7%	8%	5%	
0,7	0,57			
0,67	0,54			
4%	5%			

Figura 2.7: Comparação dos valores e seus respectivos erros relativos percentuais para o intervalo de queima de 106,61 dias efetivos de plena potência.

Fonte: Elaboração própria.

1,2				
1,09				
9%				
1,17	1,2			
1,10	1,12			
6%	7%			
1,19	1,2	1,18		
1,14	1,2	1,16		
4%	0%	2%		
1,18	1,17	1,18	1,18	
1,21	1,15	1,20	1,10	
3%	2%	2%	7%	
1,12	1,16	1,04	0,99	0,65
1,14	1,18	1,07	1,05	0,68
2%	2%	3%	6%	5%
1,02	1	0,86	0,59	
1,07	1,01	0,87	0,61	
5%	1%	1%	3%	
0,7	0,57			
0,72	0,59			
3%	4%			

Figura 2.8: Comparação dos valores e seus respectivos erros relativos percentuais para o intervalo de queima de 159,91 dias efetivos de plena potência.

Fonte: Elaboração própria.

1,19				
1,10				
8%				
1,16	1,19			
1,08	1,10			
7%	8%			
1,19	1,22	1,18		
1,12	1,16	1,11		
6%	5%	6%		
1,2	1,16	1,19	1,06	
1,18	1,11	1,13	1,03	
2%	4%	5%	3%	
1,11	1,14	1,03	0,99	0,66
1,10	1,12	1,00	0,98	0,64
1%	2%	3%	1%	3%
1,03	0,99	0,85	0,59	
1,03	0,96	0,81	0,57	
0%	3%	5%	3%	
0,7	0,57			
0,69	0,56			
1%	2%			

Figura 2.9: Comparação dos valores e seus respectivos erros relativos percentuais para o intervalo de queima de 213,22 dias efetivos de plena potência.
Fonte: Elaboração própria.

A tabela 2.3 apresenta os erros relativos máximos entre o valor presente no manual de operação do ciclo 1 de Angra-1 e a simulação, referentes para cada instante de queima.

Tabela 2.3: Erros relativos máximos.
Fonte: Elaboração própria.

Instante de queima (dias)	Erro relativo (%)
53,30	9
106,61	8
159,91	9
213,22	7

A curva de concentração de boro do modelo simulado, apresentou comportamento similar ao da curva presente no manual de operação. O SERPENT (Leppänen *et al*. 2015) possui em sua literatura, comandos que permitem a obtenção da concentração do boro no sistema em cada instante de queima do reator, um artificio facilitador no processo de

ajuste do código. A Figura 2.10 demonstra o comportamento da concentração de boro em função da queima combustível.

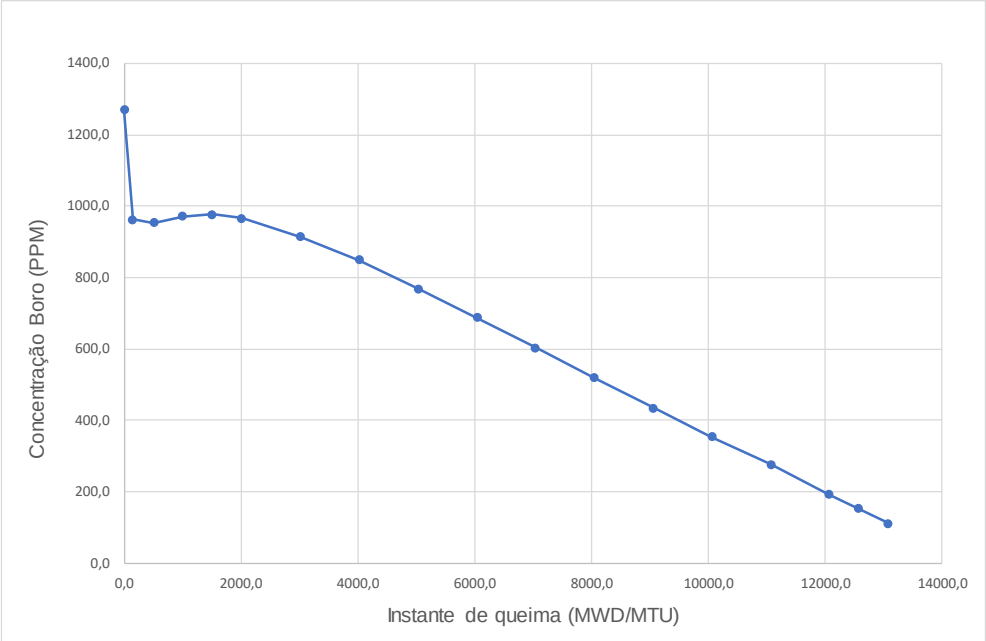

Figura 2.10: Curva do boro em relação a queima.

Fonte: Elaboração própria.

Assim, dispondo dos dados de potência referentes aos intervalos citados acima, é criada a base de dados utilizada para o treinamento da rede neural artificial e, consequentemente, a reconstrução e previsão das potências.

3. Redes Neurais Artificiais

Uma rede neural artificial (RNA), em sua forma mais genérica, pode ser descrita como uma máquina projetada para simular a maneira que o cérebro realiza uma determinada tarefa ou uma função de interesse. Dessa forma, são projetadas com intuito de imitar o sistema nervoso no processo de aprendizagem, que pode ser caracterizado pelo processamento de informações realizado pelos neurônios no cérebro, os quais são maciçamente interconectados uns com os outro (Granatyr, 2021).

Com isso, em relação ao processamento de informações pelo cérebro humano, Haykin (2009) registra que tal processo é realizado completamente diferente de um computador digital, realizando tarefas consideradas simples que um computador digital não possui capacidade de realizar.

Portanto, a fim de inferir tal capacidade aos meios de computação, foi concebida a ideia de algoritmos computacionais, que apresentam um modelo matemático que possui intuito de reproduzir a maneira que o cérebro processa informações, mesmo que de uma forma simplificada. Tais algoritmos são as redes neurais artificiais (Spörl, 2011).

Desta forma, Fleck *et al* (2016), afirma que a utilização das RNAs na resolução de problemas complexos torna-se cada vez mais uma opção atrativa, sendo, geralmente, utilizadas em problemas onde o comportamento das variáveis não é completamente conhecido.

A concepção das redes neurais artificiais, iniciou-se a partir do trabalho Warren McCulloch e Water Pitts (1943), que de acordo com Fleck *et al* (2016) apresentaram o modelo matemático de um neurônio biológico, nomeado neurônio MCP em homenagem aos seus criadores. Tal modelo matemático é caracterizado por um conjunto de *n* entradas, que são multiplicadas pela sinapse (peso), e, em seguida, os resultados são somados e submetidos a uma função que determina o valor da saída ao comparar o resultado com um descriminante, se o resultado for maior o valor da saída é igual a 1 caso contrário o valor é zero, a Figura 3.1 retrata o neurônio artificial de McCulloch e Pitts.

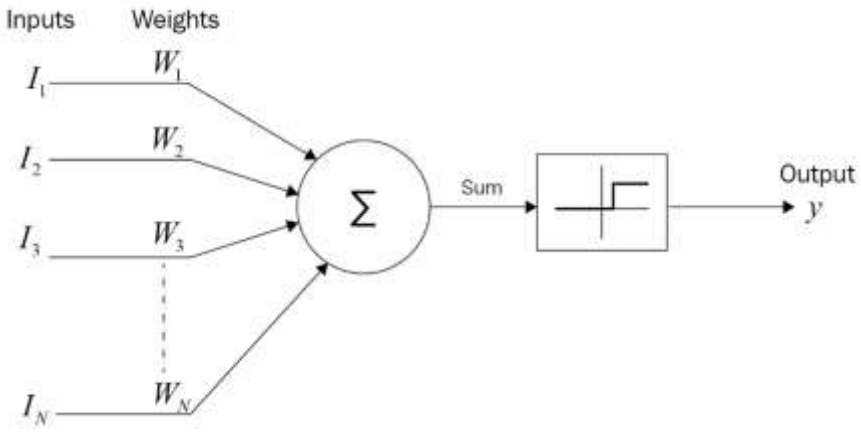

Figura 3.1: Neurônio artificial de McCulloch e Pitts.

Fonte: https://www.oreilly.com/library/view/artificial-intelligence-by/9781788990547/97eeab76-9e0e-4f41-87dc-03a65c3efec3.xhtml.

A respeito disso, em 1958, Frank Rosenblatt propõe um modelo de topologia de rede neural a partir do neurônio artificial *perceptron*, que é baseado nos neurônios propostos por McCulloch e Pitts (MCP). Tal modelo, mostrou capacidade de classificar determinados padrões ao possuir sinapses (pesos), ajustáveis (Braga e Col, 2000), a Figura 3.2 representa o neurônio artificial proposto.

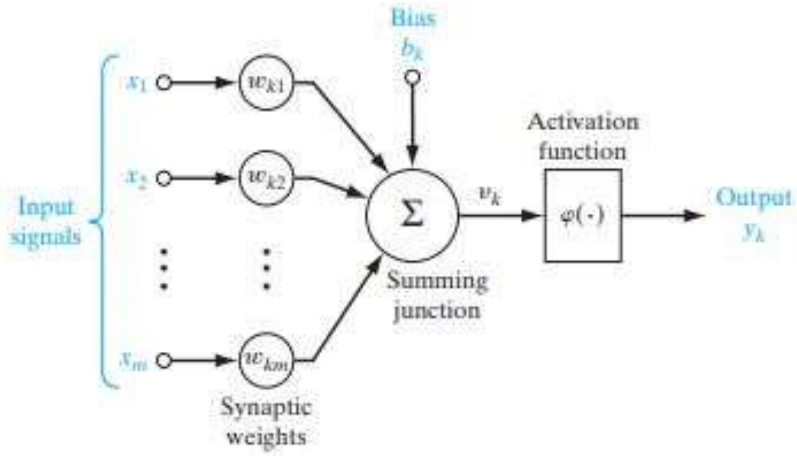

Figura 3.2: Neurônio artificial proposto por Rosenblatt.

Fonte: Haykin (2008).

Entretanto, Braga e Col (2000) registram que, devido a deficiências tecnológicas da época, era impossível a confecção de modelos físicos dos nodos e suas conexões.

Assim, estudos e pesquisas na área das RNAs sofreram um declínio, sendo retomados apenas em 1982.

Buscando compreender os algoritmos que formam as redes neurais artificiais, se faz necessária a introdução do modelo matemático de um neurônio artificial, descrito da seguinte maneira:

$$u_k = \sum_{j=1}^{m}(x_j w_j) + b_k \qquad (3.1)$$

$$y_k = \varphi(u_k) \qquad (3.2)$$

Onde os sinais de entrada são o conjunto (x_1, x_2, ..., x_m) e os pesos são o conjunto (w_1, w_2, ..., w_m) do neurônio k, onde o u_k é o resultado da multiplicação do conjunto das entradas com o conjunto dos pesos, o b_k é o bias, que ao ser empregado no neurônio, realiza o aumento ou a diminuição da entrada no neurônio (Haykin, 2008), e, por fim, y_k representa a saída do neurônio, que depende da função de ativação φ responsável por limitar o resultado da saída do neurônio a um valor fixo.

Em relação as funções de ativação, Ceccon (2020) expõe que as funções de ativação são responsáveis por atribuir as redes neurais artificiais a capacidade de processar informações não-lineares. Dentre as funções de ativação, as descritas abaixo são as mais populares (Fleck *et al*, 2016):

- Função de Limiar (Step Function): A função de limiar limita os valores de saída do neurônio da rede em valores binários (0,1). A saída assume o valor de zero quando o resultado u_k for negativo, e valor 1 quando resultado for maior do que zero, a Figura 3.3 demonstra o gráfico da função de limiar, que é expressa matematicamente da seguinte forma:

$$f(k) = \begin{cases} 1 \; se \; k \geq 0 \\ 0 \; se \; k < 0 \end{cases}$$

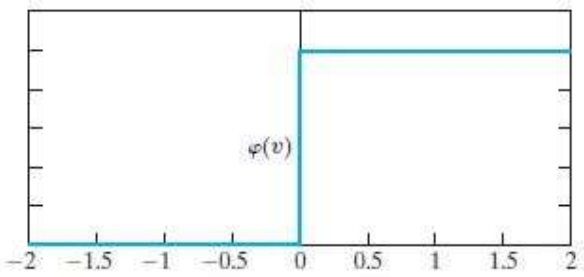

Figura 3.3: Gráfico da função de Limiar.

Fonte: Haykin (2009).

- Função sigmoide: A função sigmoide é a função de ativação mais utilizada em RNAs, de acordo com Haykin (2009), a função sigmoide é uma função estritamente crescente, que exibe um ótimo balanceamento entre o comportamento linear e não-linear, encontrando resultados no intervalo de [0,1]. A Figura 3.4 apresenta o gráfico da função sigmoide, a qual é representada pela equação:

$$y = \frac{1}{1 + e^{-u_k}}$$

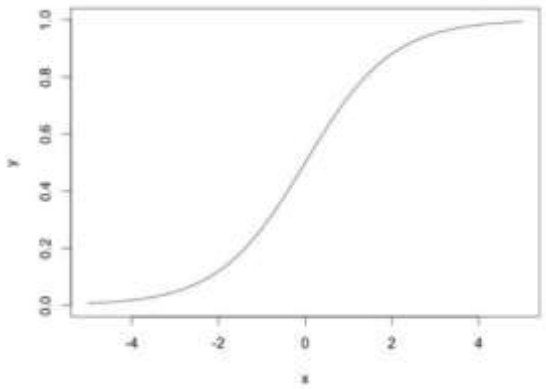

Figura 3.4: Gráfico da função sigmoide.

Fonte: http://www2.decom.ufop.br/imobilis/redes-neurais-funcoes-de-ativacao/.

- Função Tangente hiperbólica: Essa função de ativação segundo Haykin (2009), é uma função do tipo sigmoide, que permite a obtenção de resultados que variam no intervalo de [-1,1], a Figura 3.5 expõe o gráfico da função tangente.

$$y = \tanh(u_k)$$

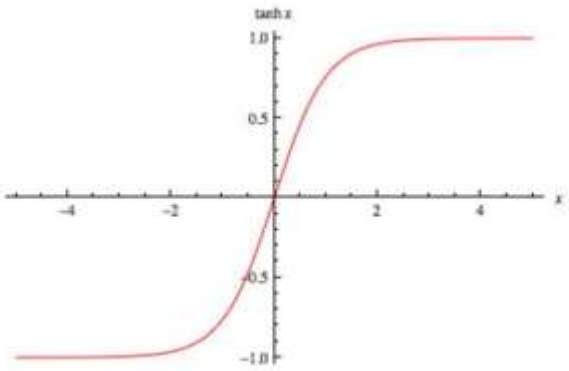

Figura 3.5: Gráfico da função Tangente.

Fonte: https://sefiks.com/2017/01/29/hyperbolic-tangent-as-neural-network-activation-function/.

A definição da arquitetura de uma rede neural artificial é um dos parâmetros mais importantes para que a RNA solucione determinado problema apresentado (Braga e Col, 2000). Ou seja, a resolução de um problema através do uso de uma rede neural artificial depende diretamente da estrutura da rede. Um exemplo disso são os nodos de McCulloch e Pitts, que são capazes de somente resolver problemas linearmente separáveis.

Com relação a isso, Haykin (2007), expõe em seu trabalho três diferentes tipos de arquiteturas de rede, que são:

- Rede *feedfoward* com camada única: Uma RNA onde seus neurônios estão organizados na forma de camadas, em sua forma mais básica, a rede é composta por uma camada com neurônios, que são conectados a uma camada dos neurônios de saída, mas não o contrário. Apesar da camada de entrada possuir neurônios, ele não realiza nenhuma computação, portanto não é contada como camada, sendo a única camada a camada de saída, a Figura 3.6 ilustra a estrutura da rede *feedfoward* de única camada.

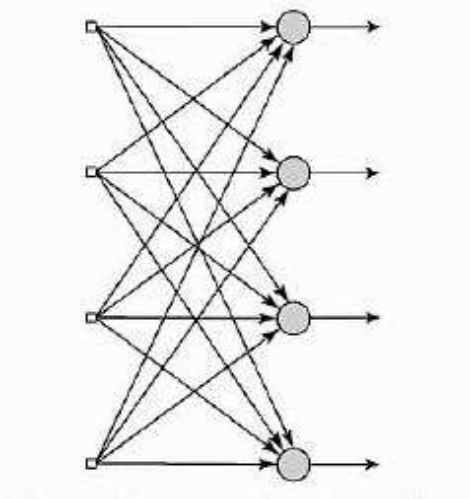

Figura 3.6: Rede feedfoward de camada única.

Fonte: Haykin (2007).

- Rede *feedfowrd* com múltiplas camadas: Uma rede do tipo *feedfoward* (Figura 3.7) é caracterizada pela presença de uma ou mais camadas ocultas, cujo nós computacionais são denominados de neurônios ocultos,

que possuem função de auxiliar a RNA no processo de encontrar os melhores resultados pro problema de interesse.

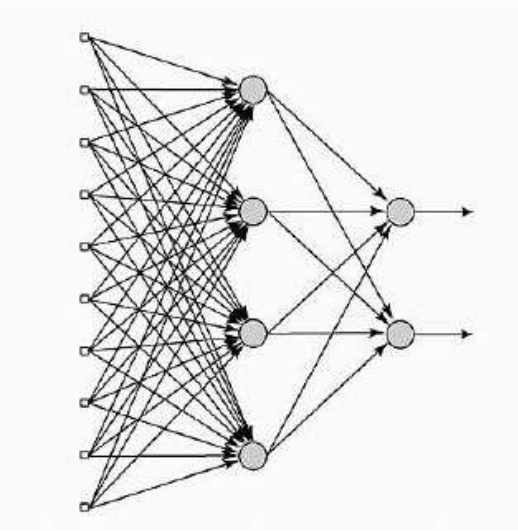

Figura 3.7: Rede *feedfoward* com múltiplas camadas.
Fonte: Haykin (2007).

- Redes recorrentes: Uma rede neural artificial, com um tipo diferente de conexão da feedfoward, onde há pelo menos um loop de realimentação na rede, um exemplo dado por Haykin (2007), é uma rede com uma única camada de neurônios com cada neurônio alimentando seu sinal de saída para as entradas dos outros neurônios da rede. A Figura 3.8 demonstra como é estruturada uma rede recorrente simples.

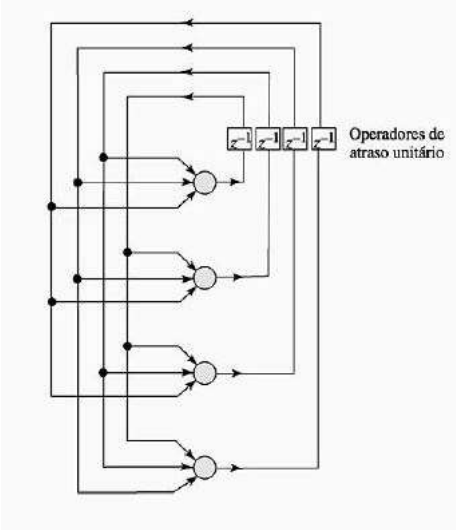

Figura 3.8: Rede recorrente.
Fonte: Haykin (2007).

Dessa forma, a topologia de rede neural artificial proposta por Rosenblatt (1958) é constituída de três camadas que são ligadas através dos pesos sinápticos, onde a primeira recebe as entradas e as repassam para a segunda camada, que é composta por neurônios que realizam as transformações nos valores empregando as funções de ativação. Por fim, a informação é enviada para a última camada, a camada de saída.

3.1 Aprendizagem

No processo de solucionar uma tarefa as redes neurais artificiais passam inicialmente por uma fase de aprendizado, na qual a rede extrai informações relevantes dos padrões das informações a ela fornecidas (Braga e Col, 2000). Assim, na etapa de aprendizagem, a RNA passa por um processo interativo de ajuste dos pesos das conexões entre os neurônios, onde, no final do processo, o conhecimento adquirido é guardado.

A cerca disso, Braga e Col (2000), esclarecem que existem diversas metodologias desenvolvidas de treinamento de redes neurais artificiais, que podem ser agrupadas em dois modelos principais: aprendizado supervisionado (com professor) e aprendizado não-supervisionado (através da experiência).

No aprendizado supervisionado é fornecido a rede os valores de saída desejados referentes aos valores de entrada, sendo viável a comparação dos valores calculados pela rede com os valores desejados (Fleck *et al*, 2016). Dessa forma, de acordo com Braga e Col (2000), tal método é denominado aprendizado supervisionado pois, tanto as entradas como as saídas desejadas são fornecidas para a RNA por um supervisor externo.

Assim, o objetivo deste método é ajustar os parâmetros da rede, no intuito de encontrar uma ligação entre os valores de entrada e de saída fornecidos. Dessa forma, a rede tem o valor de saída calculado comparado com o valor de saída desejado, recebendo informações do supervisor sobre o erro da reposta atual. Portanto, a cada entrada submetida à rede, a resposta desejada é comparada com a resposta calculada, tendo seus pesos ajustados a fim de minimizar os valores dos erros (Braga e Col, 2000).

O aprendizado supervisionado pode ser implementado de duas maneiras, *offline* ou *online*, sendo a principal diferença entre eles, a alteração dos dados de treinamento, uma vez que no aprendizado *offline* os dados de treinamento não mudam, e quando alcançada a solução pela rede, esta permanece fixa, assim, no caso de alterações na base de dados de treinamento, é necessário a realização de um novo treinamento com a nova

base. Já no aprendizado supervisionado *online*, o conjunto de dados de treinamento muda constantemente, com isso a rede permanece em processo de adaptação frequente.

A respeito do aprendizado não-supervisionado, como o próprio nome sugere, não existe um supervisor no processo de aprendizagem. Desta forma, Fleck *et al* (2016) elucida que, para alcançar uma solução, a rede deve procurar algum tipo de correlação ou redundância nos dados de entrada.

Logo, este tipo de método somente é possível quando houver uma redundância nos dados de entrada, uma vez que a rede somente apresenta solução após estabelecer um equilíbrio estatístico dos dados.

O treinamento da RNA utilizada no desenvolvimento deste trabalho, dá-se por meio do aprendizado supervisionado, onde as entradas fornecidas a rede têm suas referentes saídas.

3.2 Rede Neural de Múltiplas camadas.

As redes neurais artificiais de múltiplas camadas, ou em inglês *Multilayer Perceptron* (MLP), é uma rede neural com a categoria *feed-foward*, na qual cada camada alimenta seus dados para a camada seguinte, sucedendo-se até a camada de saída. Em seu treinamento, uma rede MLP, realiza o ajuste dos pesos e bias dos neurônios artificiais para que a resposta desejada para o problema seja encontrada.

A rede do tipo MLP tem suas camadas compostas por neurônios artificiais *perceptron* propostos por Rosenblatt (1958), tendo em sua estrutura pelo menos uma camada escondida, podendo conter múltiplas entradas e múltiplas saídas, a Figura 3.9 apresenta a configuração de uma rede do tipo MLP.

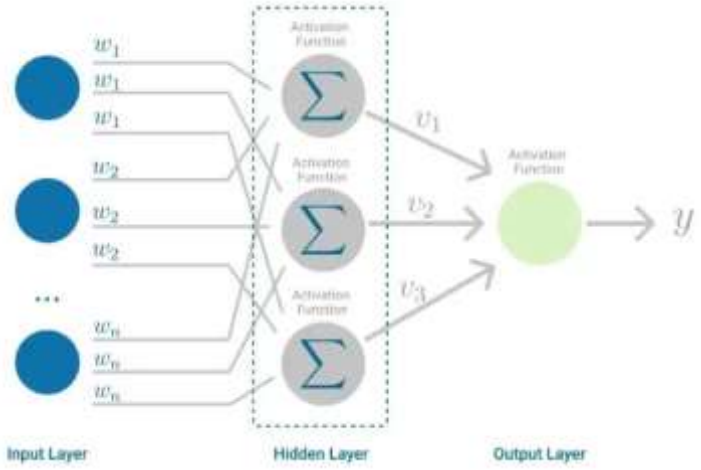

Figura 3.9: Rede neural artificial MLP.

Fonte: https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-

life-example-and-python-code-sentiment-analysis-cb408ee93141.

Desta maneira, Moita (2012) aponta em sua pesquisa que um dos modelos mais conhecidos de redes neurais artificiais é o modelo *Multilayer Perceptron Backpropagation*, que utiliza como algoritmo de treinamento o algoritmo *Backpropagation*. Utilizado amplamente para solução de diferentes problemas, tais como classificação de padrões (reconhecimento), controle e processamento de sinais (Miguez, 2012)

Com este algoritmo, quando um conjunto de entradas é inserido a rede é produzida inicialmente uma saída, em seguida é medida a distância entre a resposta da saída atual e a desejada, e logo após os ajustes nos pesos são feitos a fim de diminuir esta distância. Este processo é conhecido como Regra Delta (Miguez, 2012).

Desta forma, o algoritmo de treinamento mais utilizado no treinamento de redes do tipo MLP, é uma generalização da Regra Delta denominado algoritmo *Backpropagation*, que define de forma automática os pesos da rede MLP, assim conferindo a capacidade de apresentar soluções de funções lineares e não lineares.

Durante o treinamento, a rede opera em uma sequência de dois passos, no primeiro passo é inserido um conjunto de entradas, os valores resultantes das funções de ativação fluem através da rede, camada por camada até a camada de saída. No segundo passo a saída calculada é comparada com a saída desejada, caso a saída não esteja correta o erro é calculado e transmitido a partir da camada de saída até a camada de entrada, assim os pesos das conexões são modificados na medida que o erro é retropropagado.

Dessa maneira, sobre a rede neural de múltiplas camadas, Hornik *et al* (1989) em seu trabalho, exprimem que tal modelo de rede, apresenta-se capaz de aproximar qualquer função mensurável em graus satisfatórios, podendo ser considerada como um aproximador universal.

Ademais, para esta pesquisa a rede neural implementada é a rede neural de múltiplas camadas, pois além da sua capacidade de aprendizado e resolução de problemas, é uma rede de fácil implementação.

Em seguida discutiremos a utilização do modelo de rede neural de múltiplas camadas para a previsão e reconstrução da densidade de potência do núcleo do reator de referência.

No intuito de realizar a reconstrução e previsão dos valores de densidade de potência do reator proposto, a rede neural artificial é estruturada de forma que seja efetuado o modelo de regressão linear.

Zhang *et al* (2019) expressam que problemas de regressão normalmente surgem quando se há o interesse em prever um valor numérico, exemplos mais comuns são a previsão de preços de casa ou ações, duração de estadia de um paciente em um hospital, entre outros. Dessa forma, a implementação da rede neural para a regressão linear, é um método viável, no qual apresenta valores satisfatórios tendo uma base de dados de treinamento suficientemente grande.

Sobre a regressão linear, Rodrigues (2012) a define como um conjunto de técnicas estatísticas usadas para formar relações entre variáveis e predizer o valor de uma ou mais variáveis dependentes. Portanto, o modelo de regressão linear consiste de uma variável dependente estar relacionada a uma ou mais variáveis independentes por uma equação linear (Medeiros e Bianchi, 2009).

A equação linear identifica a melhor reta que se ajusta aos dados baseada no método dos mínimos quadrados (Filho, 2002). Desta forma, na tentativa de estabelecer a equação que representa o evento estudado é possível efetuar a construção de um gráfico, denominado de diagrama de dispersão (Figura 3.10), onde é verificado como a variável dependente (y) comporta-se em função da variável independente (x) (Peternelli, s.d.). Assim, o Método dos Mínimos Quadrados pode ser caracterizado num método que visa a obtenção de uma equação que as distancias entre os pontos no diagrama e da reta sejam as mínimas possíveis.

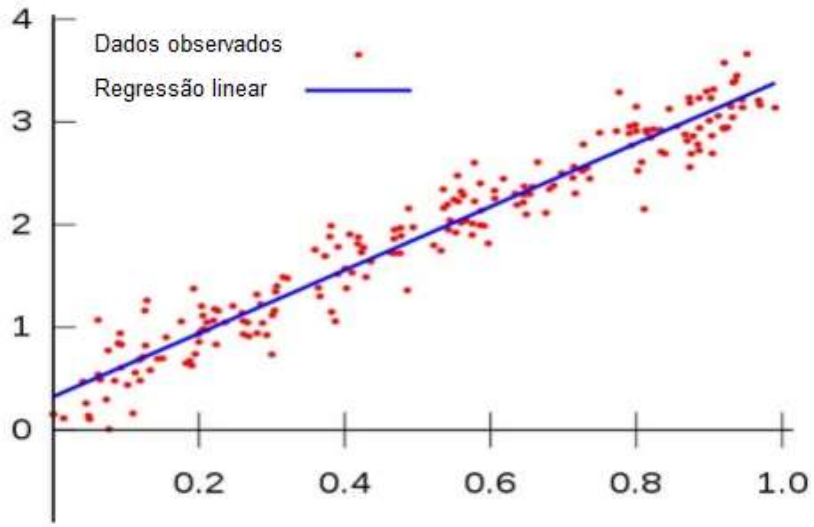

Figura 3.10: Diagrama de dispersão e reta ajustante.

Fonte: https://pt.venngage.com/blog/diagrama-de-dispersao/.

Para o modelo de regressão com rede neural artificial a equação que prediz os valores com uma determinada base de dados, pode ser expressa por:

$$\hat{y} = w_1 x_1 + \ldots + w_i x_i + b \qquad (3.3)$$

Onde o w é a denotação do peso referente a determinada entrada, e o b é o bias.

Portanto, na tentativa de adequar o modelo da rede aos valores da base de dados, são necessários alguns parâmetros que sirvam para medir o desempenho do modelo, a função erro é um dos parâmetros mais importantes nesse quesito (Zhang *et al*, 2019). A função erro mede a distância entre os dados reais e os dados previstos. Uma representação dessa distância é demonstrada na Figura 3.11.

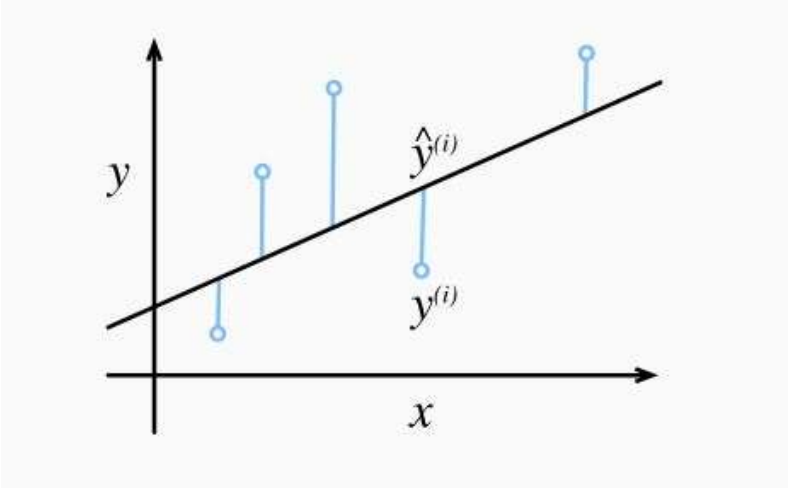

Figura 3.11: Erro no modelo de regressão.

Fonte: Zhang *et al* (2019).

A função erro que é mais utilizada em problemas deste tipo, é a função do quadrado das diferenças, podendo ser representada por:

$$(\hat{y}^{(i)} - y^{(i)})^2 \qquad (3.4)$$

Entretanto, no aprendizado de máquinas, normalmente, trabalha-se com uma ampla quantidade de dados, portanto o erro é expresso da seguinte forma:

$$E = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}^{(i)} - y^{(i)})^2 \qquad (3.5)$$

Assim, na implementação de uma rede neural artificial, é imprescindível a implementação de uma função que calcule o erro, que é uma função que tem como papel comparar a saída desejada com o valor da saída encontrada, determinando o quão bem a rede neural lida com a base de dados de treinamento.

Portanto, no treinamento da rede, é almejado a minimização dos erros entre o valor desejado e o valor alcançado. Dessa forma, a função de erro implementada na rede neural artificial, empregada nesta pesquisa, é a função média da soma dos erros ao quadrado (3.4).

Dessa maneira, no processo da propagação da rede, o erro é calculado pela função escolhida, e, no final, seguindo o algoritmo de treinamento *Backpropagation*, a rede é retropropagada, a fim de otimizar o erro por meio da alteração dos parâmetros da rede.

Na retropropagação da rede é aplicado o algoritmo de otimização chamado gradiente descendente, o qual, segundo Zhang *et al* (2019), é uma técnica de otimização essencial para quase qualquer tipo de modelo de aprendizagem de máquina. Tal método encontra os valores dos parâmetros que minimizam o erro de forma interativa. Usando o conceito de gradiente negativo, o algoritmo minimiza a função no qual ele é aplicado, na direção da descida mais íngreme (Vaz, 2020).

Para melhor visualização do método é possível partir do modelo de regressão simples, onde a reta ajustante é definida por $y(x) = ax + b$ com seu erro calculado pelo somatório do quadrado das diferenças.

Faz-se necessário utilizar o gradiente descendente para encontrar os melhores parâmetros, e, dessa forma, minimizar o erro entre o valor encontrado e o desejado. Para isso é aplicado o conceito de derivada parcial na função que calcula o erro, uma vez que quanto mais próximo de zero for a derivada da função mais próximo encontra-se o ponto mínimo.

Dessa forma, para o ajuste dos parâmetros temos:

$$a = a - \beta * \frac{\partial erro}{\partial a} \qquad (3.6)$$

$$b = b - \beta * \frac{\partial erro}{\partial b} \qquad (3.7)$$

Para a correção dos parâmetros é dado um chute inicial dos valores a e b da reta, que, em seguida, serão otimizados interativamente. O parâmetro β é a taxa de aprendizagem do algoritmo, que será responsável pela distância percorrida em cada

interação. Uma taxa de aprendizagem muito baixa leva a um grande custo computacional, uma vez que o número de interações até atingir o ponto mínimo da função aumenta consideravelmente, já com uma taxa muito alta, o algoritmo pode falhar em alcançar o ponto mínimo uma vez que pode passar por ele (Vaz, 2020). A Figura 3.12 demonstra o processo do gradiente descendente.

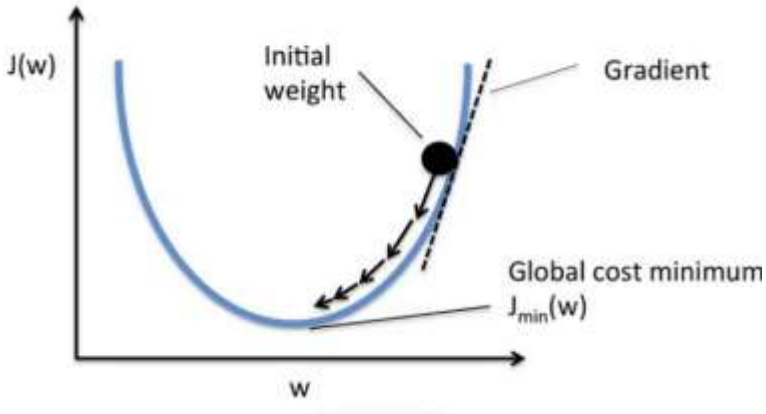

Figura 3.12: Gradiente descendente.

Fonte: Vaz, 2020. Disponível em: https://medium.com/data-hackers/gradientes-descendentes-na-pr%C3%A1tica-melhor-jeito-de-entender-740ef4ff6c43#:~:text=A%20gradiente%20descedente%20%C3%A9%20um,os%20par%C3%A2metros%20do%20nosso%20modelo

4.Metodologia e Implementação

Seguindo a metodologia descrita no capítulo anterior, é construída uma RNA para a reconstrução das densidades do tipo *feedfoward* de múltiplas camadas, contendo 3 camadas ocultas para melhor computação dos dados. Onde os seus valores de entrada são referentes aos 36 valores fornecidos pelos detectores internos, e terá 121 valores de saídas, referentes aos valores de potência de todos os elementos combustíveis do núcleo do reator. A Figura 4.1 apresenta o funcionamento da rede neural artificial do tipo *feedfoward* de múltiplas camadas.

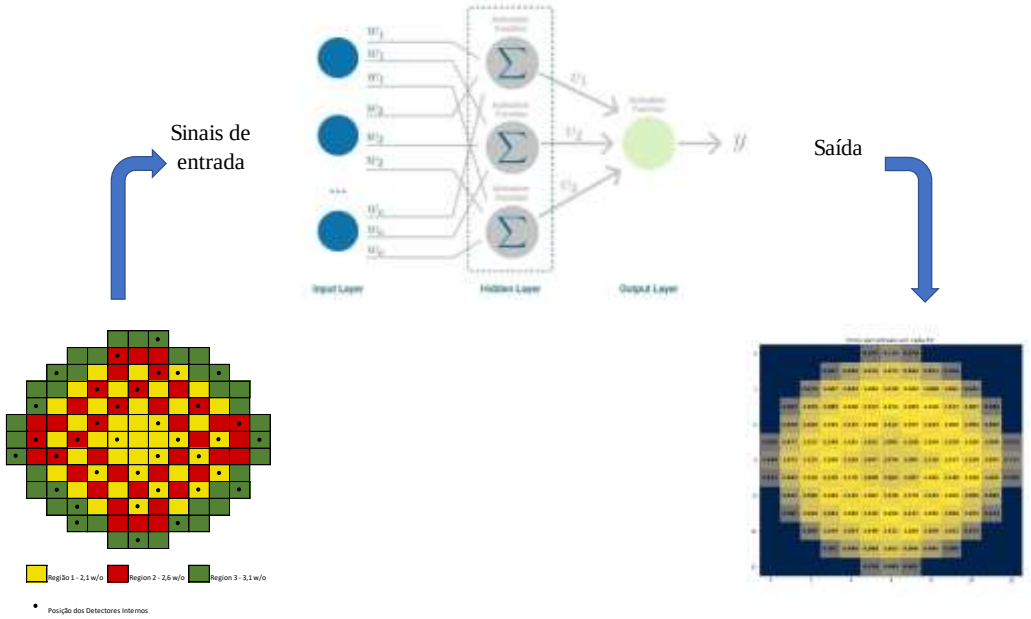

Figura 4.1: Diagrama ilustrando funcionamento da RNA de múltiplas camadas do tipo *feedfoward*.

Fonte: Elaboração própria.

Dessa forma, como o aprendizado da rede é dado da forma de aprendizado supervisionado, a rede será alimentada como valores de entrada os valores dos 36 detectores internos em determinado instante de queima, e como valores de saída os valores de potência dos 121 elementos combustíveis no mesmo instante de queima.

A RNA, foi treinada utilizando duas bases de dados, no intuito de permitir análise do desempenho da rede e verificação da implementação da metodologia.

Logo, na primeira base de dados, são utilizados 14 instantes de queima para o treinamento e 4 instantes para a avaliação de desempenho, na seguinte disposição:

- Treinamento: 0 - 3,99 – 26,65 – 39,98 – 53,30 – 106,61 – 133,26 – 186,57 – 239,87 – 266,52 – 293,18 – 319,83 – 333,16 - 346,48 (dias efetivos a plena potência).
- Avaliação: 13,33 – 79,96 – 159,91 – 213,22 (dias efetivos a plena potência).

Na segunda base de dados, são empregados 5 instantes de queima, com um considerável intervalo entre eles, e um instante de queima para avaliação do desempenho da rede:

- Treinamento: 0 – 26,65 – 53,30 – 106,61 – 159,91 (dias efetivos a plena potência.
- Avaliação: 3,99 (dias efetivos a plena potência).

Por fim para estudo da viabilidade da metodologia, a RNA é treinada com dois diferentes conjuntos de entradas, o primeiro conjunto de entradas é referente aos 36 detectores internos do SDNI, e no segundo conjunto, são utilizados os valores de resposta de apenas 33 detectores internos, dessa forma no segundo conjunto a rede terá apenas 33 entradas.

O treinamento e implementação da RNA, é realizado por meio da linguagem Python (1991), com o auxilio da biblioteca de código aberto TensorFlow (2015), desenvolvida pelo Google. A biblioteca TensorFlow (2015), auxilia no processo de aprendizagem de maquinas, onde disponibiliza APIs que facilitam a criação de modelos. A palavra API segundo Melo (2021), é um acrônimo para as palavras inglesas *application programming interface*, sendo um termo usado para definir uma *interface* de comunicação que um sistema oferece para que outros acessem suas funções e recursos. O API utilizado para a elaboração da metodologia é o Keras (2015), qual foi incorporado pela biblioteca TensorFlow em 2017.

Ademais, nos treinamentos da RNA, são variados dois parâmetros, que são: algoritmo de treinamento e função de ativação dos neurônios das camadas ocultas. Assim para os algoritmos de treinamento, são implementados o SGD (*Stochastic Gradient Descent)* que é o método clássico do gradiente descendente, e o outro algoritmo utilizado é o Adam, que segundo a literatura Keras, é um algoritmo de gradiente descendente que utiliza do método do momento.

Segundo a literatura, o método Adam é computacionalmente mais eficiente e possui pouco uso de memória, sendo apropriado para problemas que possuem uma vasta quantidade de dados.

Já, no que diz a respeito as funções de ativação, são usadas as funções: tangente hiperbólica e a função sigmoide.

Dessa forma, para cada base de dados, a rede passara por 4 treinamentos diferentes, onde a arquitetura da rede em cada treinamento será:

- Rede MPL com algoritmo de treinamento Adam e função de ativação tangente hiperbólica;
- Rede MPL com algoritmo de treinamento Adam e função de ativação sigmoide;
- Rede MPL com algoritmo de treinamento SGD e função de ativação tangente hiperbólica;
- Rede MPL com algoritmo de treinamento SGD e função de ativação sigmoide.

5. Resultados

Neste trabalho foram simuladas as respostas dos detectores do SDNI do reator PWR similar ao de Angra-1. Em seguida os valores simulados foram usados para formar a base de dados para o treinamento da RNA responsável pela reconstrução e previsão da distribuição de densidade de potência do núcleo do reator.

Dessa forma, as respostas dos detectores do SDNI foram obtidas por meio da metodologia apresentada no capítulo 2, no qual utilizadas como valores de entradas para a rede neural artificial apresentada nos capítulos 3 e 4, realizou a previsão e reconstrução da distribuição de densidade de potência.

A fim de obter dados suficientes para gerar uma base de dados que viabilize o treinamento da rede, foram utilizados os valores de 18 instantes de queimas simulados, demonstradas na Figura 5.1.

Tabela 5.1: Instantes de queima e seus dias equivalentes.

Fonte: Elaboração própria.

Instantes de queima simulados	Equivalência em dias
1	0
2	3,99
3	13,33
4	26,65
5	39,98
6	53,3
7	79,96
8	106,61
9	133,26
10	159,91
11	186,57
12	213,22
13	239,87
14	266,52
15	293,18
16	319,83
17	333,16
18	346,48

Além disso a rede neural artificial implementada possui 3 camadas de neurônios ocultos sendo constituídas de 72, 108 e 140 neurônios respectivamente. Tal estruturação visa otimizar o processo de computação das entradas pela rede, onde por meio de testes constatou-se incapaz de fornecer valores razoáveis com um baixo número de neurônios nas camadas ocultas.

Portanto, neste capítulo são apresentados os resultados do método empregado nesta pesquisa.

5.1 Reconstrução com respostas de 36 detectores internos.

Foram utilizadas as respostas referentes aos 36 detectores internos do núcleo do reator, com qual foi construída uma base de dados para os treinamentos da rede que será implementada para a realização da reconstrução da densidade de potência.

Nos treinamentos a rede recebe 36 entradas que são referentes aos detectores do SDNI. Assim foram realizados dois treinamentos da rede com as diferentes bases de dados expostas no capítulo 4, para melhor validação da metodologia.

Os treinamentos como seus resultados são descritos nos subcapítulos subsequentes.

5.1.1 Primeira base de dados.

Nos treinamentos da primeira base de dados, são usados para treinamento e avaliação, os valores dos seguintes dias efetivos a plena potência:

- Treinamento: 0 - 3,99 – 26,65 – 39,98 – 53,30 – 106,61 – 133,26 – 186,57 – 239,87 – 266,52 – 293,18 – 319,83 – 333,16 - 346,48 (dias efetivos a plena potência).
- Avaliação: 13,33 – 79,96 – 159,91 – 213,22 (dias efetivos a plena potência).

Assim, a rede neural artificial passa por 4 treinamentos onde são empregadas as 4 arquiteturas apresentadas no capítulo 4. Os erros relativos percentuais para cada elemento combustível (EC) entre os valores previstos pela RNA e os valores simulados pelo SERPENT (valores desejados), são expressos nas Figuras 5.1, 5.2, 5.3 e 5.4. É necessário registrar que, para melhor avaliação da RNA, é definido como admissíveis erros relativos de no máximo de 10%.

Assim, nas duas primeiras figuras os erros são correspondentes ao instante de queima de 13,33 dias com diferentes funções de ativação onde o mesmo foi aplicada para as duas últimas figuras que correspondem ao instante de queima 159,91 dias onde o algoritmo de treinamento usado foi o Adam.

O erro foi calculado da seguinte maneira:

$$E = \frac{|V_d - V_p|}{V_d} \text{x} 100 \qquad (5.1)$$

Onde:

E: Erro relativo percentual;

V_d: Valor a ser encontrado desejado;

V_p: Valor previsto pela rede neural artificial.

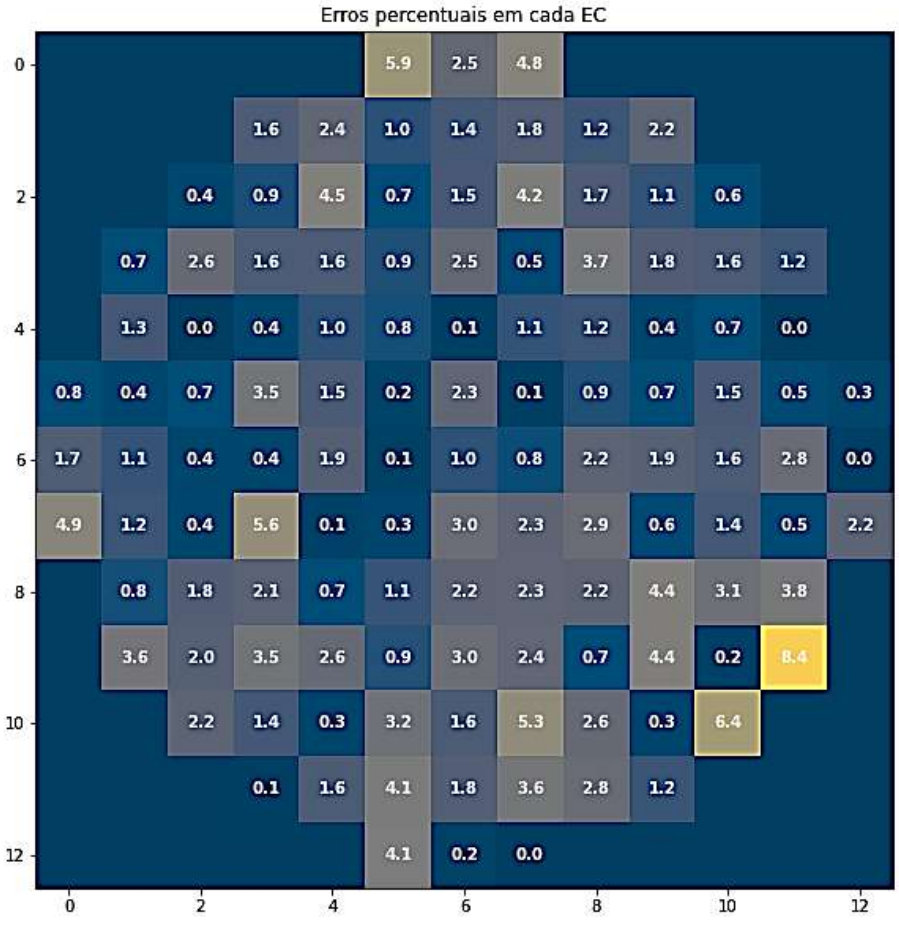

Figura 5.1: Erros percentuais para cada EC no instante de queima 13,33 dias com função de ativação tangente hiperbólica.

Fonte: Elaboração própria.

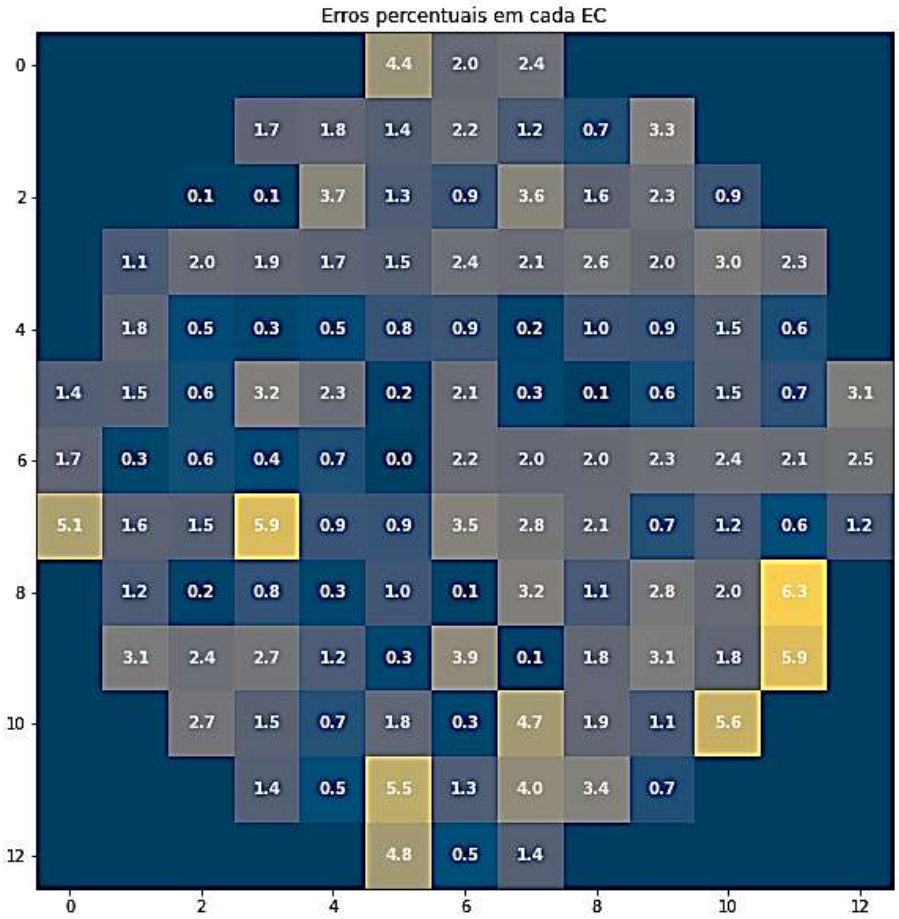

Figura 5.2: Erros percentuais para cada EC no instante de queima 13,33 dias com função de ativação sigmoide.

Fonte: Elaboração própria.

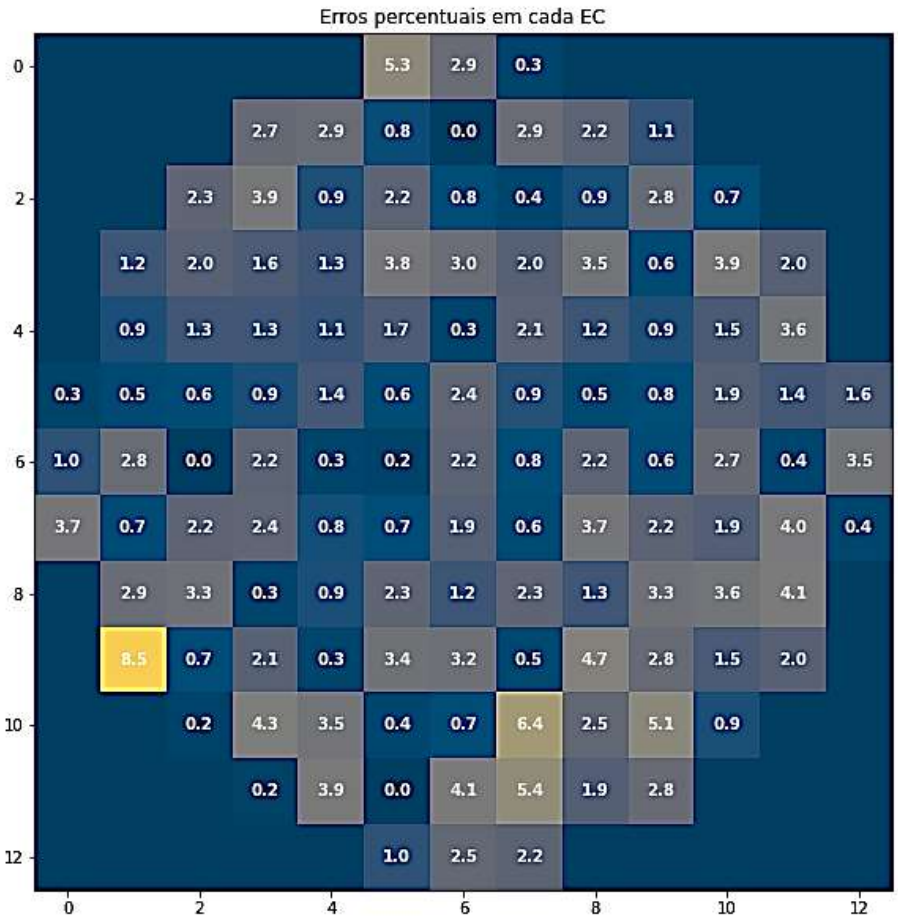

Figura 5.3: Erros percentuais para cada EC no instante de queima 159,91 dias com função de ativação tangente hiperbólica.

Fonte: Elaboração própria.

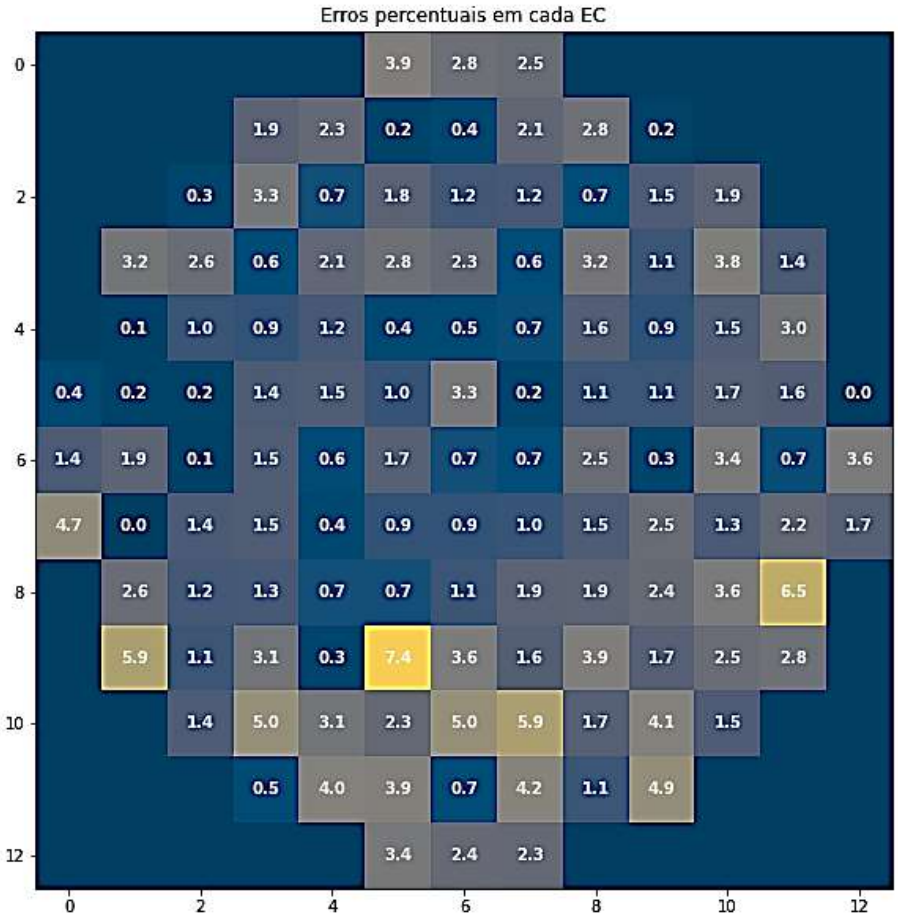

Figura 5.4: Erros percentuais para cada EC no instante de queima 159,91 dias com função de ativação sigmoide.

Fonte: Elaboração própria.

Nos treinamentos onde foi empregado o algoritmo de treinamento Adam, é alcançado valores razoáveis na distribuição de erro, onde o maior erro apresenta valor de 8,5%, e também ao observar os resultados é possível perceber que a rede que tem como função sigmoide como função de ativação dos neurônios das camadas ocultas, apresenta melhor distribuição dos erros e menor erro máximo que a função tangente hiperbólica.

Em seguida são apresentados pelas Figuras 5.5, 5.6, 5.7 e 5.8 os erros dos elementos combustíveis no primeiro treinamento com a implementação do algoritmo de treinamento SGD.

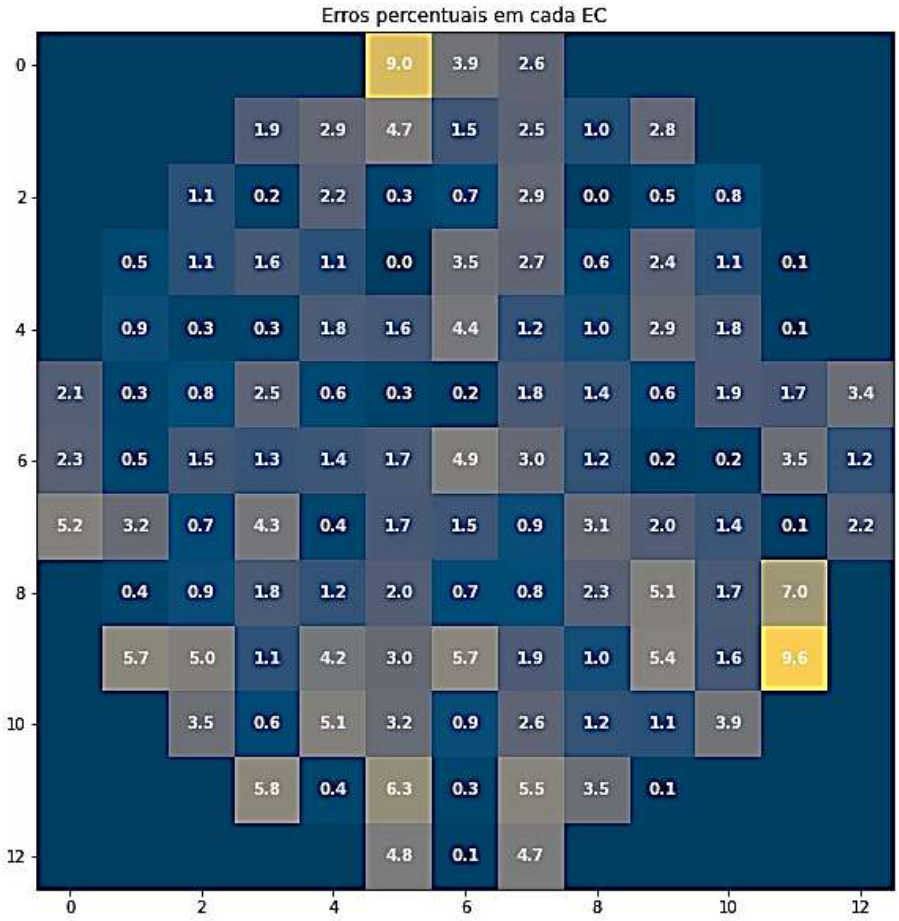

Figura 5.5: Erros percentuais para cada EC no instante de queima 13,33 dias com função de ativação tangente hiperbólica.

Fonte: Elaboração própria.

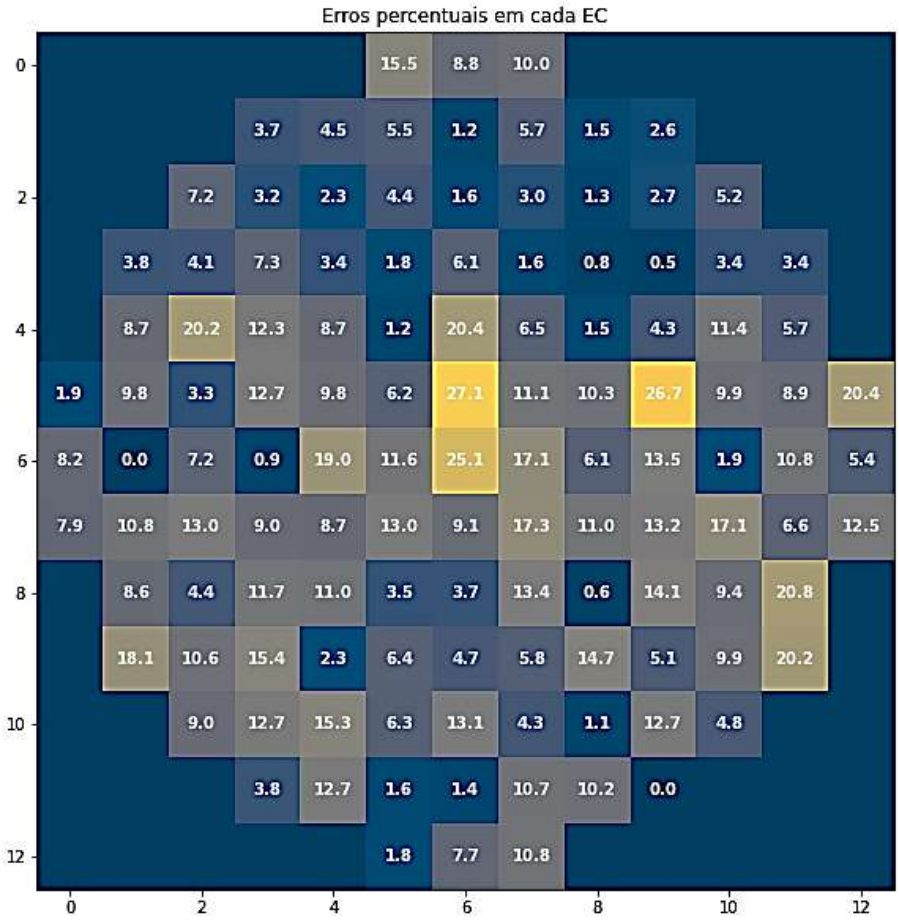

Figura 5.6: Erros percentuais para cada EC no instante de queima 13,33 dias com função de ativação sigmoide.

Fonte: Elaboração própria.

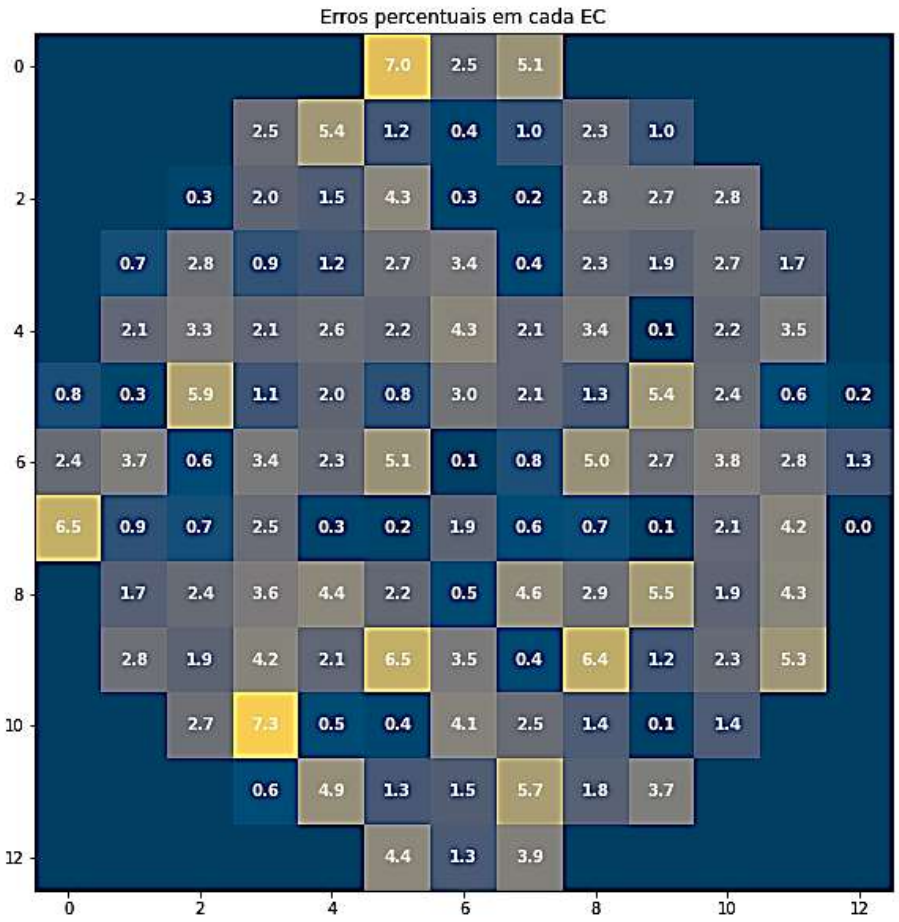

Figura 5.7: Erros percentuais para cada EC no instante de queima 159,91 dias com função de ativação tangente hiperbólica.

Fonte: Elaboração própria.

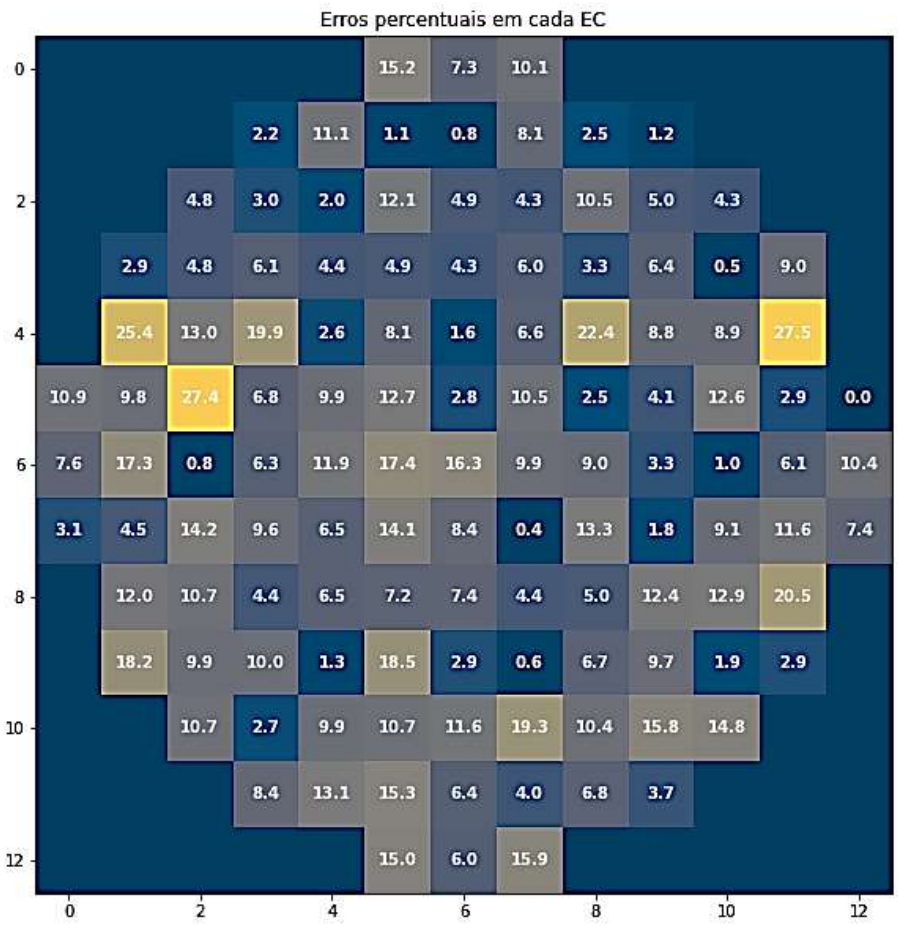

Figura 5.8: Erros percentuais para cada EC no instante de queima 159,91 dias com função de ativação sigmoide.

Fonte: Elaboração própria.

Analisando os erros é possível perceber que quando a rede é treinada por meio da utilização do algoritmo de treinamento Adam, os valores percentuais dos erros em sua maioria, apresentam-se razoáveis. Uma vez que, no elemento combustível que apresenta maior erro percentual, esse erro não ultrapassa os 10%, conforme as Figuras 5.5 e 5.7. Já, o algoritmo de treinamento SGD não apresentou bons resultados quando é utilizado em conjunto com a função de ativação sigmoide, apresentando erros próximos a 30%, vide as Figuras 5.6 e 5.8. A Tabela 5.2 apresenta os valores dos erros relativos máximos na primeira base de dados em todas configurações de redes realizadas.

Tabela 5.2: Erros relativos máximos da primeira base de dados.

Fonte: Elaboração própria.

Erros relativos máximos				
	Adam		SGD	
Instante de queima	Sigmoide	Tangente hiperbolica	Sigmoide	Tangente hiperbolica
13,33	6,3	8,4	27,1	9,6
159,91	7,4	8,5	27,5	7,3

5.1.2 Segunda base de dados.

Na segunda base de dados, os valores para treinamento e validação da RNA usados são referentes aos seguintes dias efetivos a plena potência:

- Treinamento: 0 – 26,65 – 53,30 – 106,61 – 159,91 (dias).
- Avaliação: 3,99 (dias).

Os 4 treinamentos implementados na primeira base de dados, foram implementados para a segunda base de dados. Assim, para a rede com algoritmo de treinamento Adam os resultados são apresentados pelas Figuras 5.9 e 5.10.

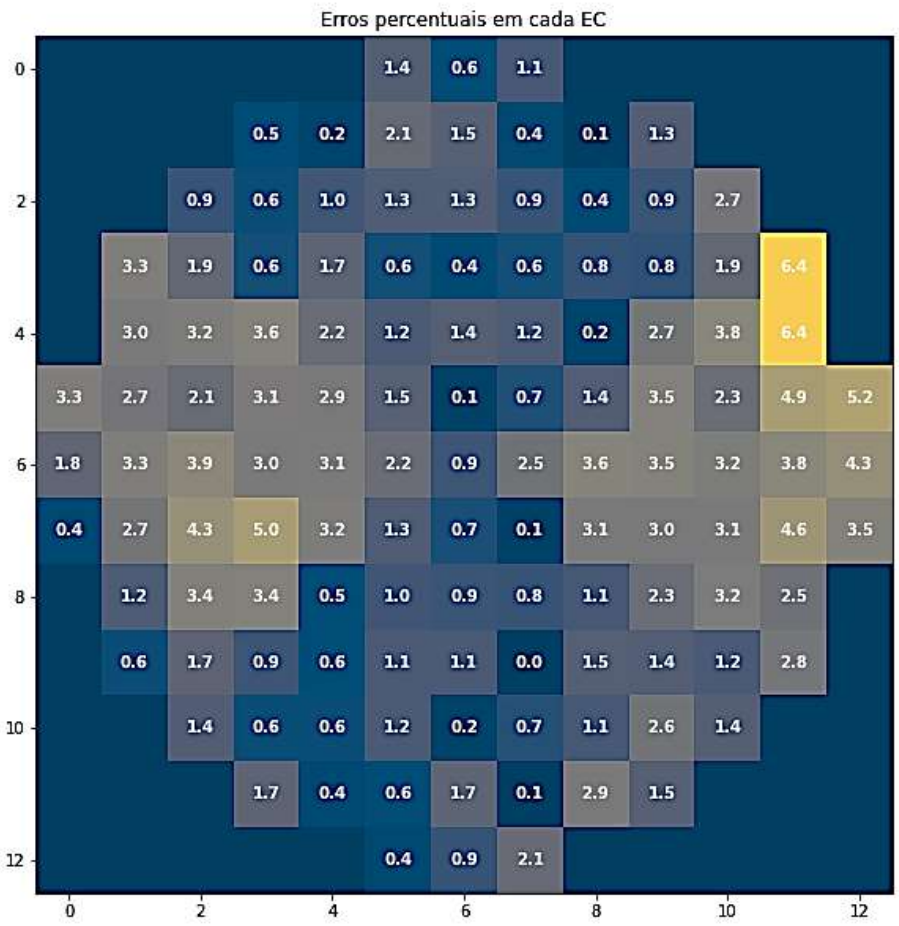

Figura 5.9: Erros percentuais para cada EC no instante de queima 3,99 dias com função de ativação tangente hiperbólica.

Fonte: Elaboração própria.

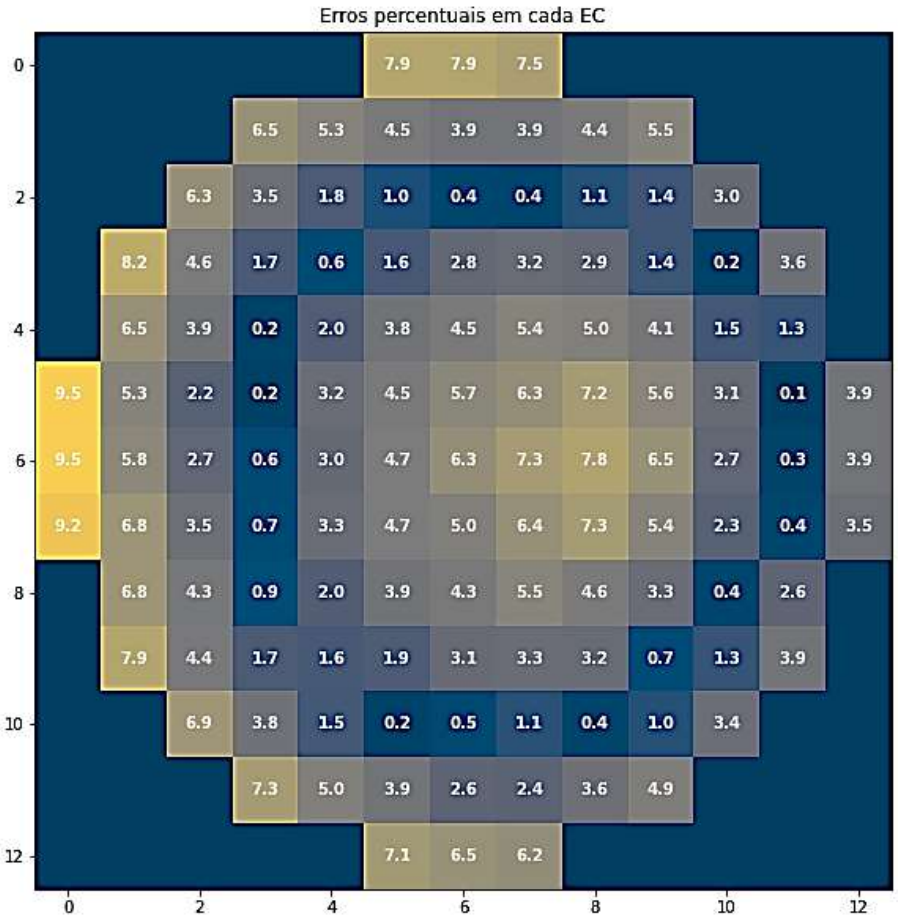

Figura 5.10: Erros percentuais para cada EC no instante de queima 3,99 dias com função de ativação sigmoide.

Fonte: Elaboração própria.

Nos treinamentos da rede com algoritmo de aprendizado Adam na segunda base de dados, a estrutura de rede com a função de ativação sigmoide, apresentou uma distribuição de erro e erro máximo maior que a estrutura com a função de ativação tangente hiperbólica, resultados distintos dos resultados dos treinamentos com a primeira base de dados.

Os resultados do treinamento com a implementação do algoritmo de treinamento SGD são representados nas Figuras 5.11 e 5.12.

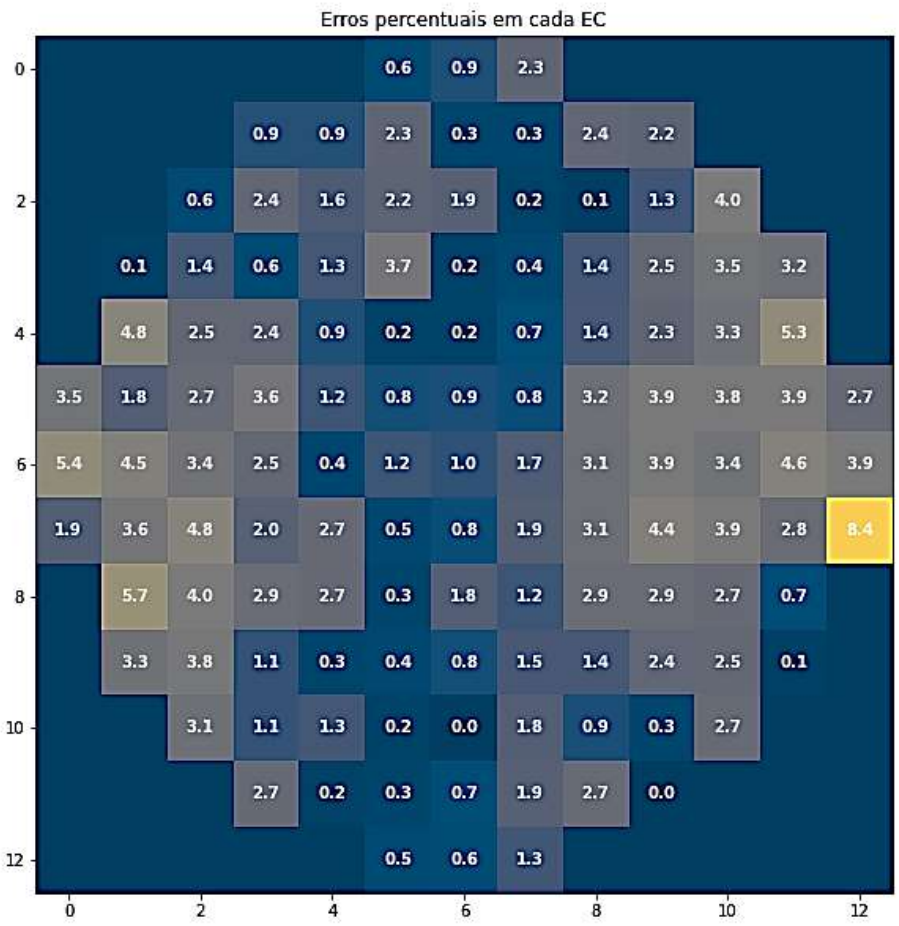

Figura 5.11: Erros percentuais para cada EC no instante de queima 3,99 dias com função de ativação tangente hiperbólica.

Fonte: Elaboração própria.

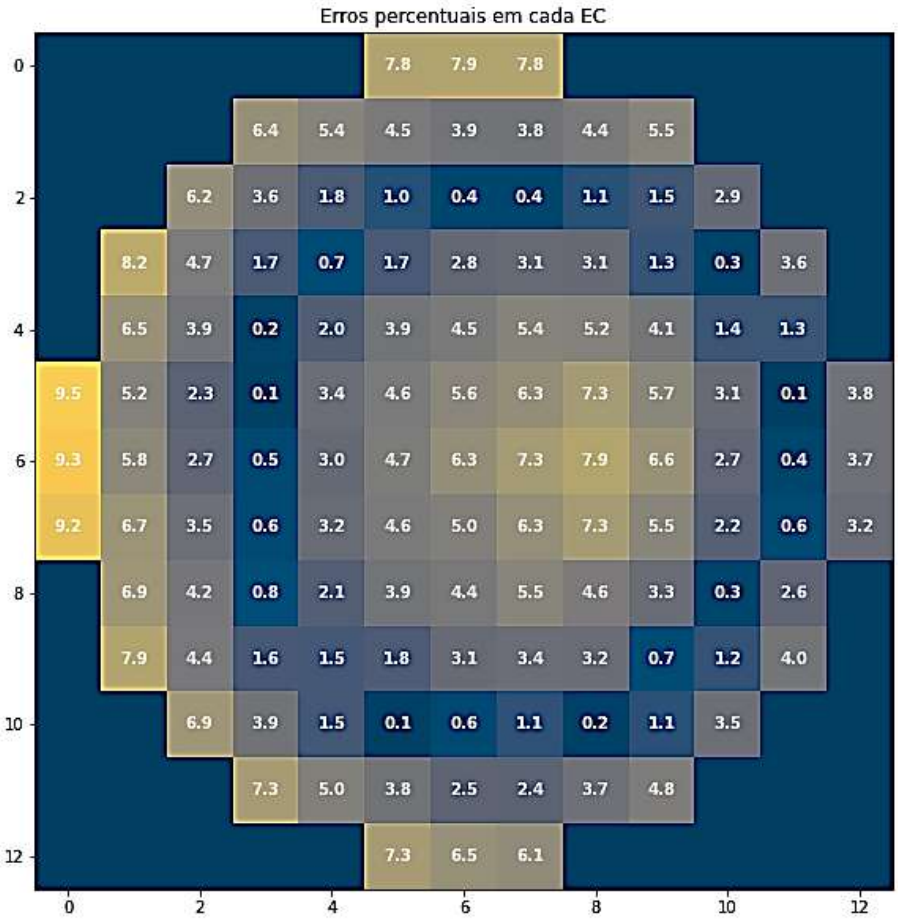

Figura 5.12: Erros percentuais para cada EC no instante de queima 3,99 dias com função de ativação sigmoide.

Fonte: Elaboração própria.

Nos resultados dos treinamentos onde a rede neural artificial tem como algoritmo de treinamento o SGD, os valores dos erros são próximos aos valores da rede com algoritmo de treinamento Adam, o mesmo não é percebido na primeira base de dados pois de acordo com a interface Keras, o algoritmo de treinamento Adam apresenta melhores resultados em problemas onde há uma grande quantidade de dados, sendo o caso da primeira base de dados. A tabela 5.3 apresenta os erros relativos máximos dos resultados obtidos para a segunda base de dados.

Tabela 5.3: Erros relativos máximos da segunda base de dados.

Fonte: Elaboração própria.

Erros relativos máximos				
	Adam		SGD	
Instante de queima	Sigmoide	Tangente hiperbolica	Sigmoide	Tangente hiperbolica
3,99	9,5	6,4	9,5	8,4

5.2 Reconstrução com respostas de 33 detectores internos.

Foram utilizadas como valores de entrada para a RNA, as respostas de 33 detectores do SDNI, as posições dos detectores internos que tiveram seus valores usados no treinamento e os detectores que não participaram do treinamento são apresentadas pela Figura 5.13.

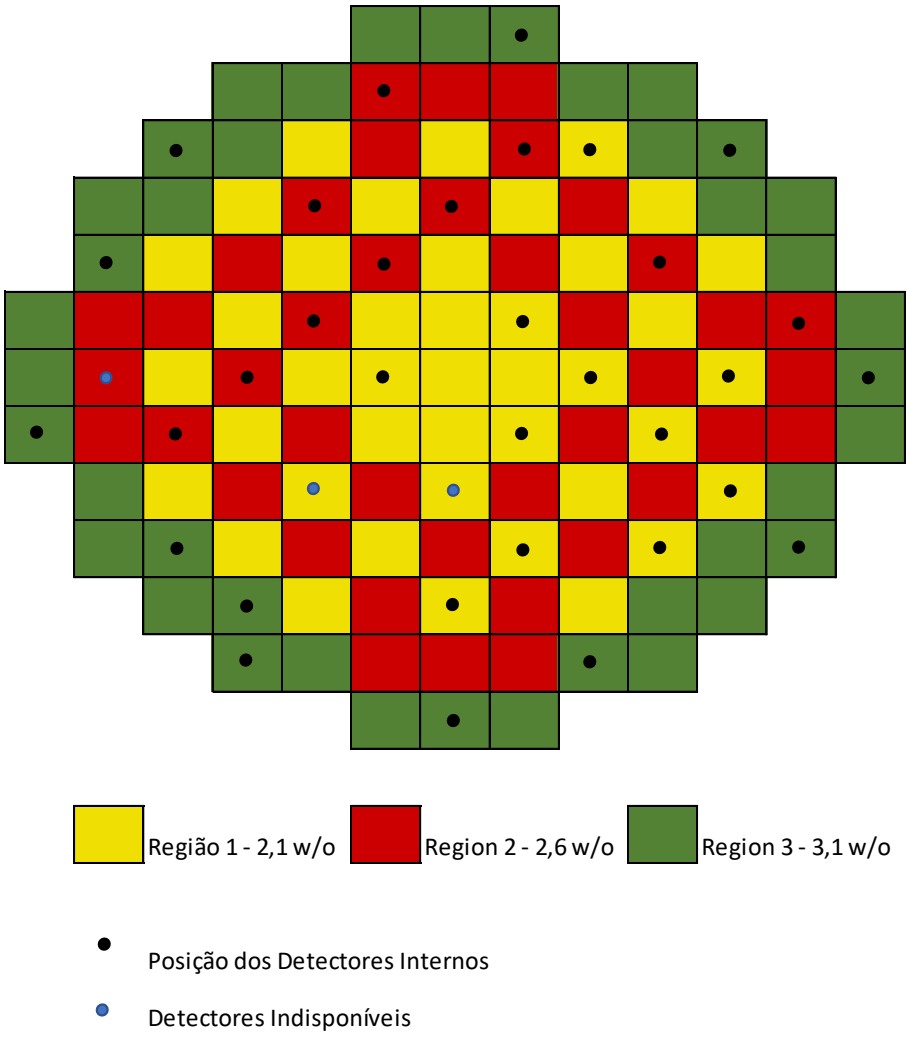

Figura 5.13: Posição dos 33 detectores utilizados no treinamento da rede.

Fonte: Elaboração própria.

5.2.1 Primeira base de dados.

Para a primeira base de dados, a RNA passou pelos mesmos treinamentos que a rede neural com 33 entradas, sendo utilizados os mesmos algoritmos de treinamento e funções de ativação.

As Figuras 5.14, 5.15, 5.16 e 5.17 retrata os valores dos resultados do primeiro treinamento da rede, onde mantêm-se inalterável o algoritmo de treinamento, que é o Adam.

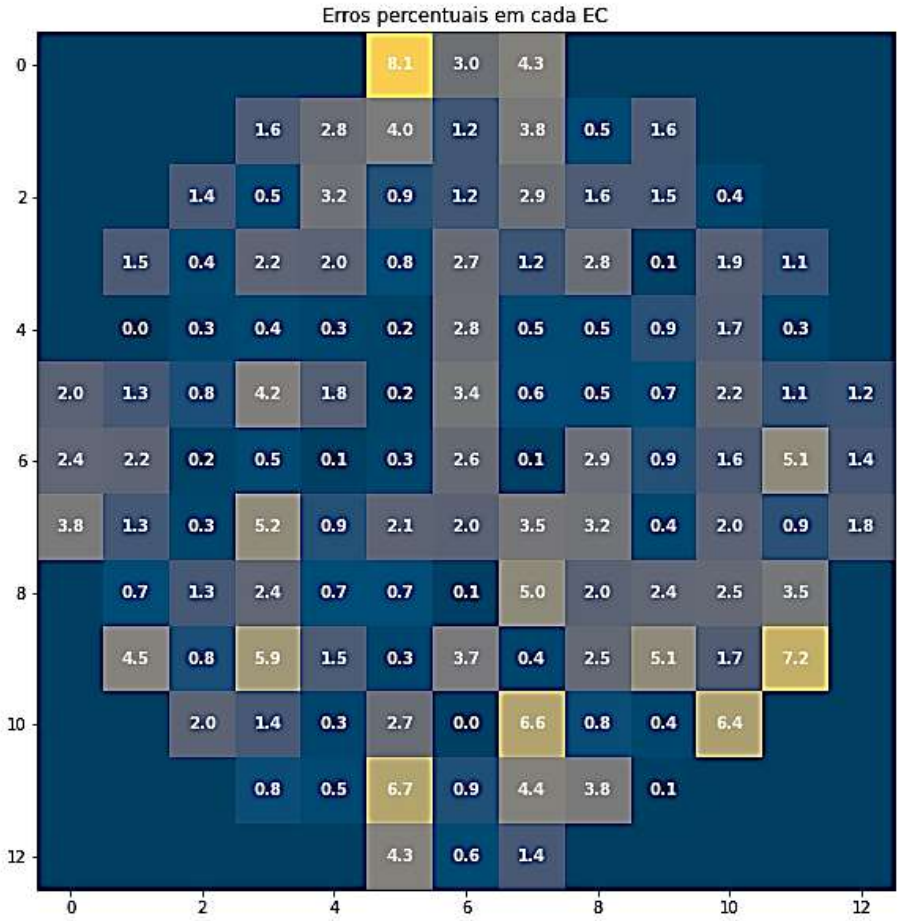

Figura 5.14: Erros percentuais para cada EC no instante de queima 13,33 dias com função de ativação tangente hiperbólica.

Fonte: Elaboração própria.

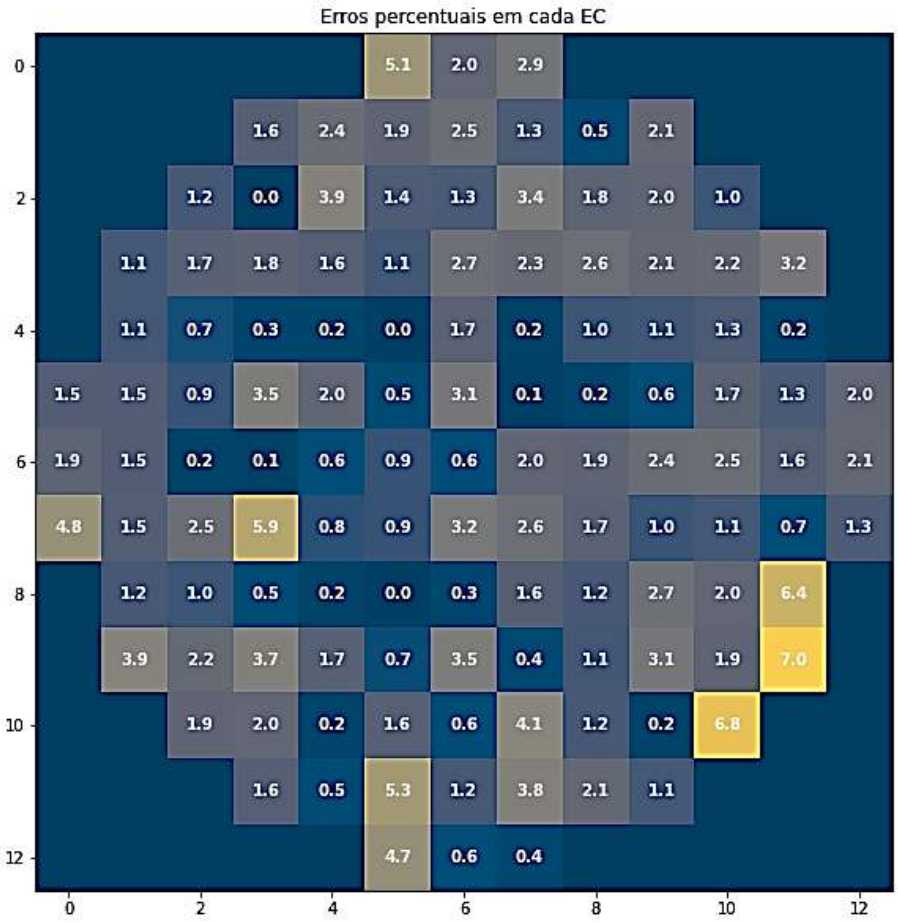

Figura 5.15: Erros percentuais para cada EC no instante de queima 13,33 dias com função de ativação sigmoide.

Fonte: Elaboração própria.

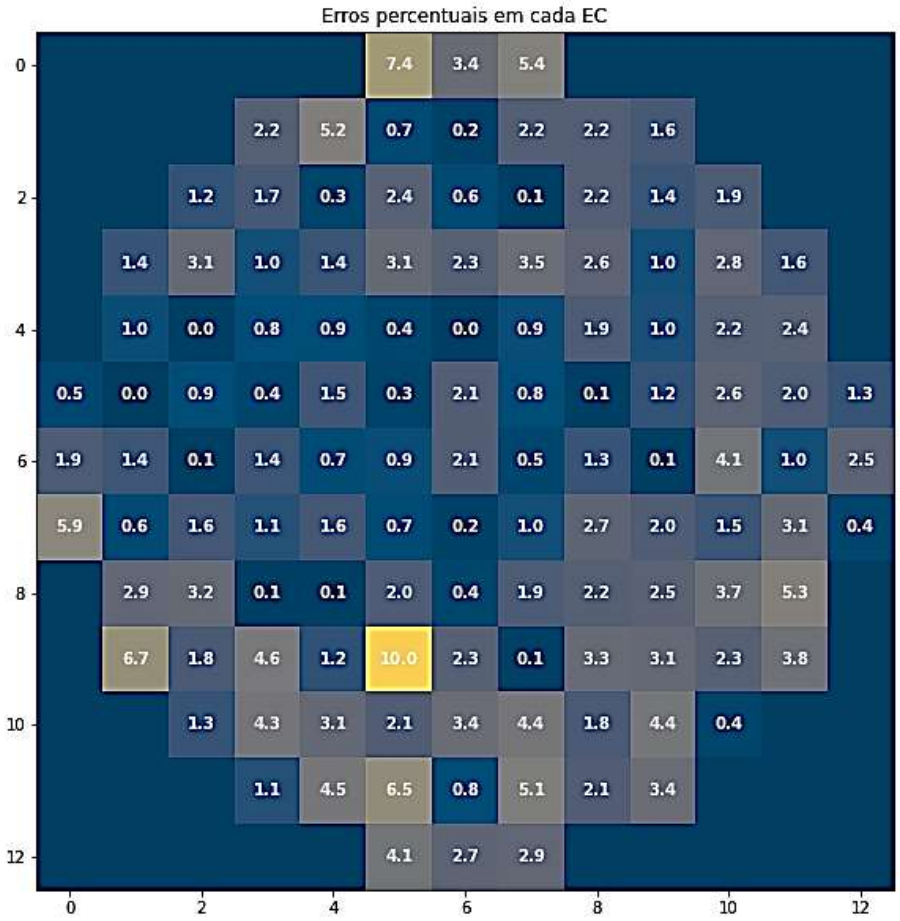

Figura 5.16: Erros percentuais para cada EC no instante de queima 159,91 dias com função de ativação tangente hiperbólica.

Fonte: Elaboração própria.

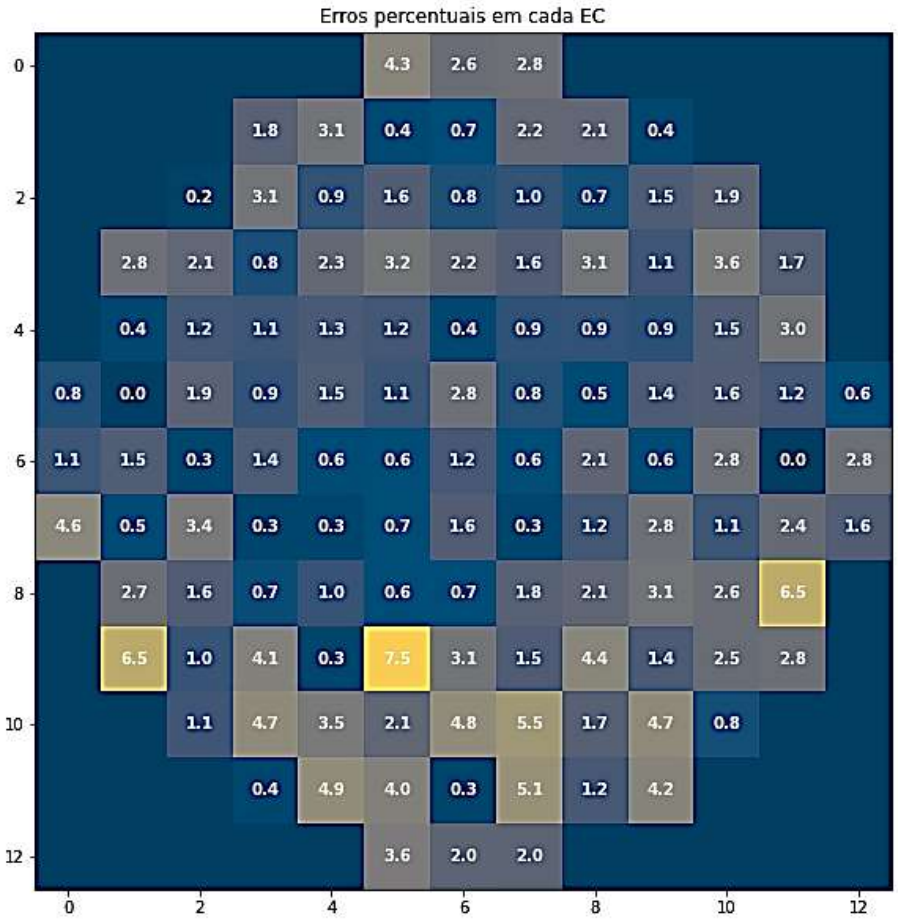

Figura 5.17: Erros percentuais para cada EC no instante de queima 159,91 dias com função de ativação sigmoide.

Fonte: Elaboração própria.

Diante desses resultados, é possível perceber a semelhança com os resultados da rede neural treinada com 33 entradas, onde os valores dos erros da rede que possui a função sigmoide nos neurônios das camadas ocultas são menores que os valores da rede com função de ativação tangente hiperbólica. Entretanto o erro máximo é maior que a rede com 36 entrada, com valor de 10%, que pode ser explicado pelo fato da rede possuir um menor número de entradas para computação dos dados, causando uma maior imprecisão na previsão dos dados.

Em seguida, são apresentados pelas figuras 5.18, 5.19, 5.20 e 5.21 os erros percentuais com a utilização do algoritmo de treinamento SGD, onde a ordem de apresentação será idêntica as dos resultados previamente apresentadas neste capítulo.

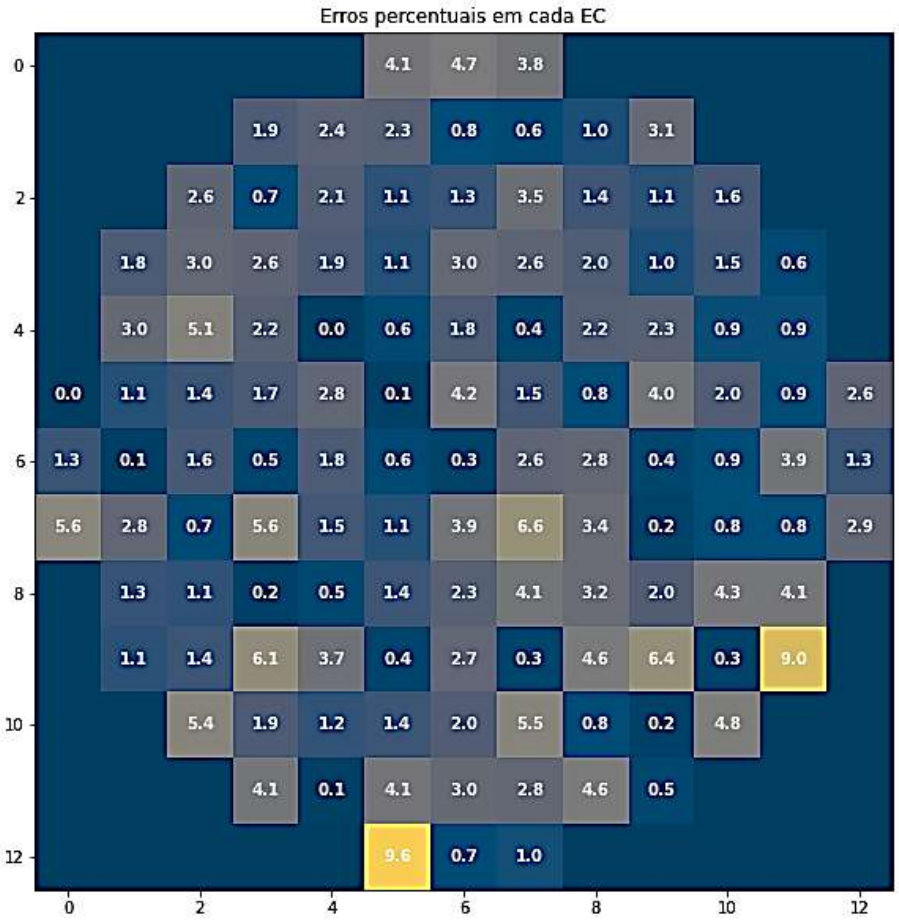

Figura 5.18: Erros percentuais para cada EC no instante de queima 13,33 dias com função de ativação tangente hiperbólica.

Fonte: Elaboração própria.

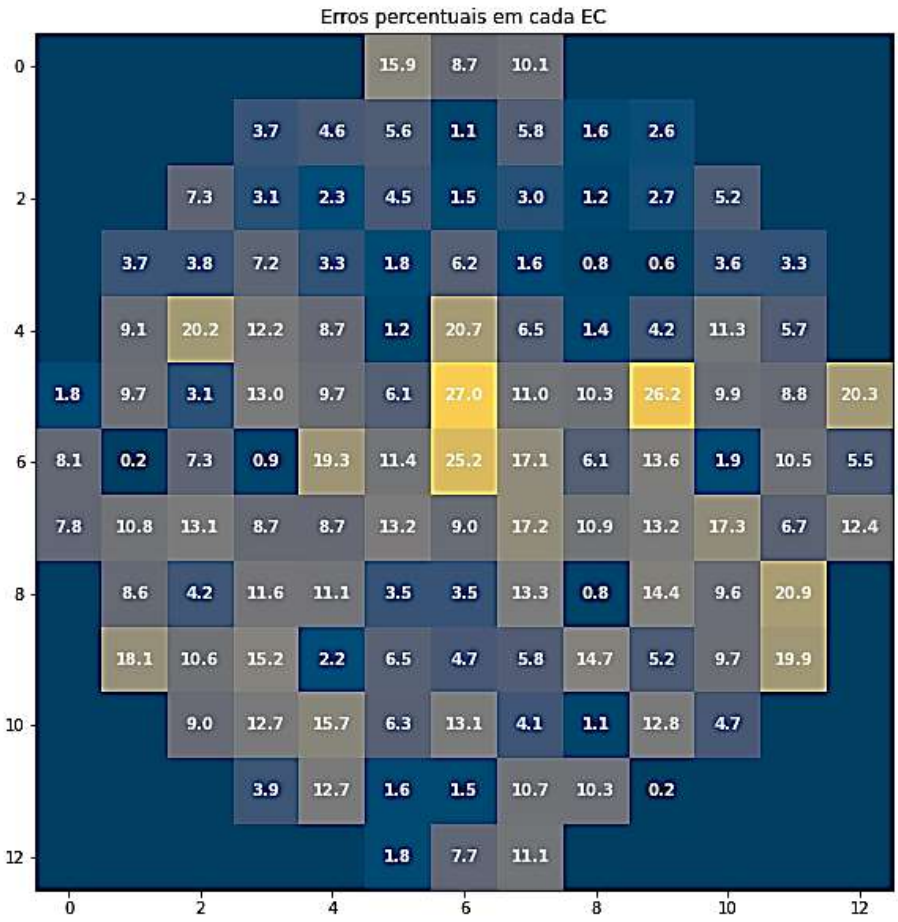

Figura 5.19: Erros percentuais para cada EC no instante de queima 13,33 dias com função de ativação sigmoide.

Fonte: Elaboração própria.

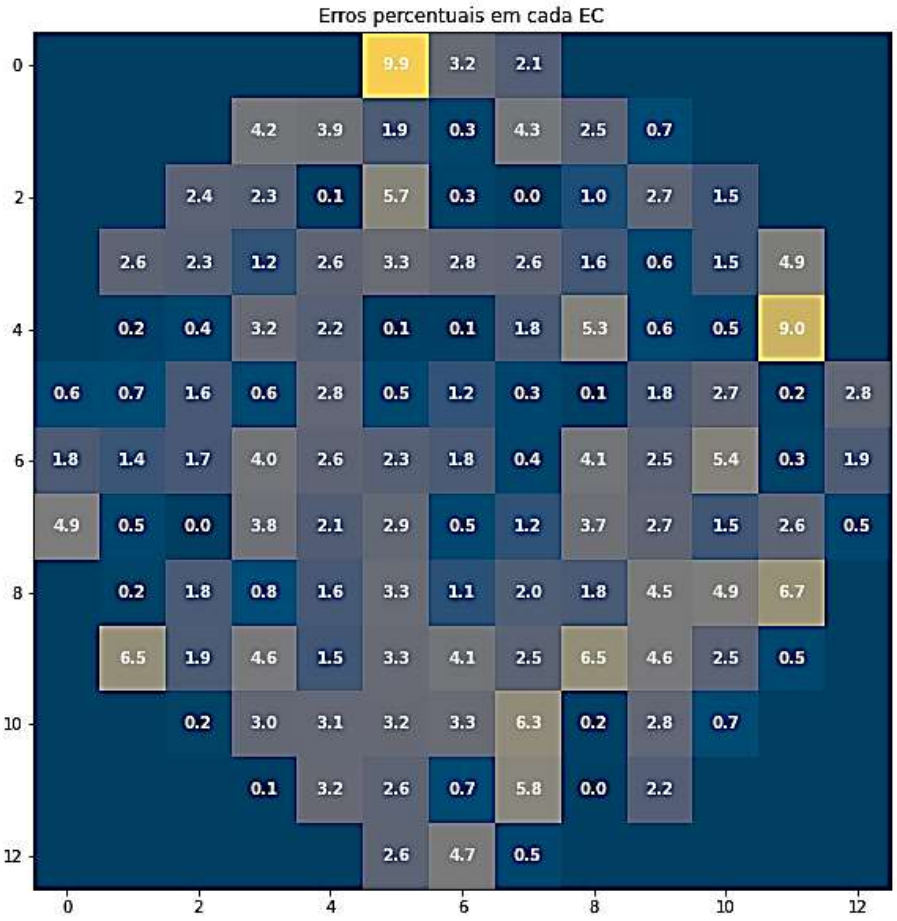

Figura 5.20: Erros percentuais para cada EC no instante de queima 159,91 dias com função de ativação tangente hiperbólica.

Fonte: Elaboração própria.

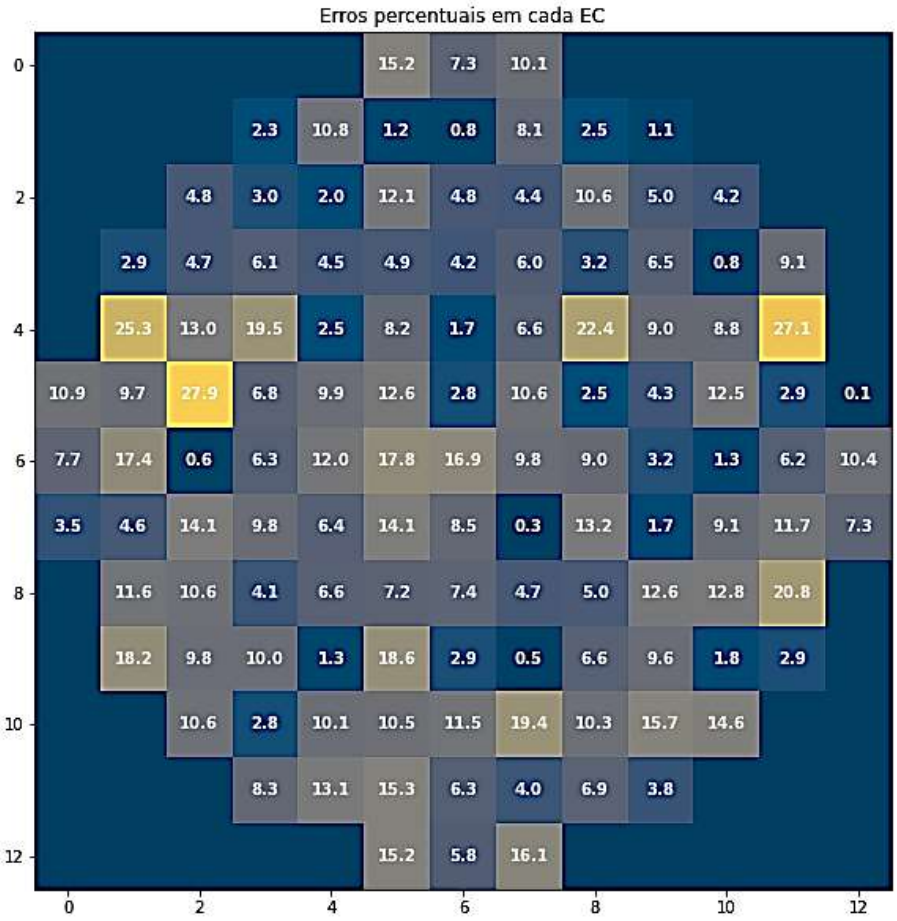

Figura 5.21: Erros percentuais para cada EC no instante de queima 159,91 dias com função de ativação sigmoide.

Fonte: Elaboração própria.

Ao observar os resultados é possível inferir que os erros, apresentam comportamento similar ao primeiro treinamento da rede que utiliza das 36 respostas dos detectores internos, onde nas estruturas da RNA onde o algoritmo de treinamento é o SGD e a função de ativação é a sigmoide, os erros são demasiadamente altos, aproximando-se dos 28%. A tabela 5.4 apresenta os erros relativos máximos dos resultados obtidos pela rede com 33 entradas na primeira base de dados.

Tabela 5.4: Erros relativos máximos da primeira base de dados.

Fonte: Elaboração própria.

Erros relativos máximos				
	Adam		SGD	
Instante de queima	Sigmoide	Tangente hiperbolica	Sigmoide	Tangente hiperbolica
13,33	7,0	8,1	27,0	9,6
159,91	7,5	10,0	27,9	9,9

5.2.2 Segunda base de dados.

Na segunda base de dados os valores, para o treinamento são utilizados os valores referentes a 5 instantes de queima, e para avaliação da rede são usados um 1 instante de queima, sendo os mesmos instantes que foram utilizados para a rede com 33 entradas.

Assim sendo, os resultados dos erros referentes aos treinamentos da rede utilizando como algoritmo de treinamento o algoritmo Adam, são demonstrados pelas Figuras 5.22 e 5.23.

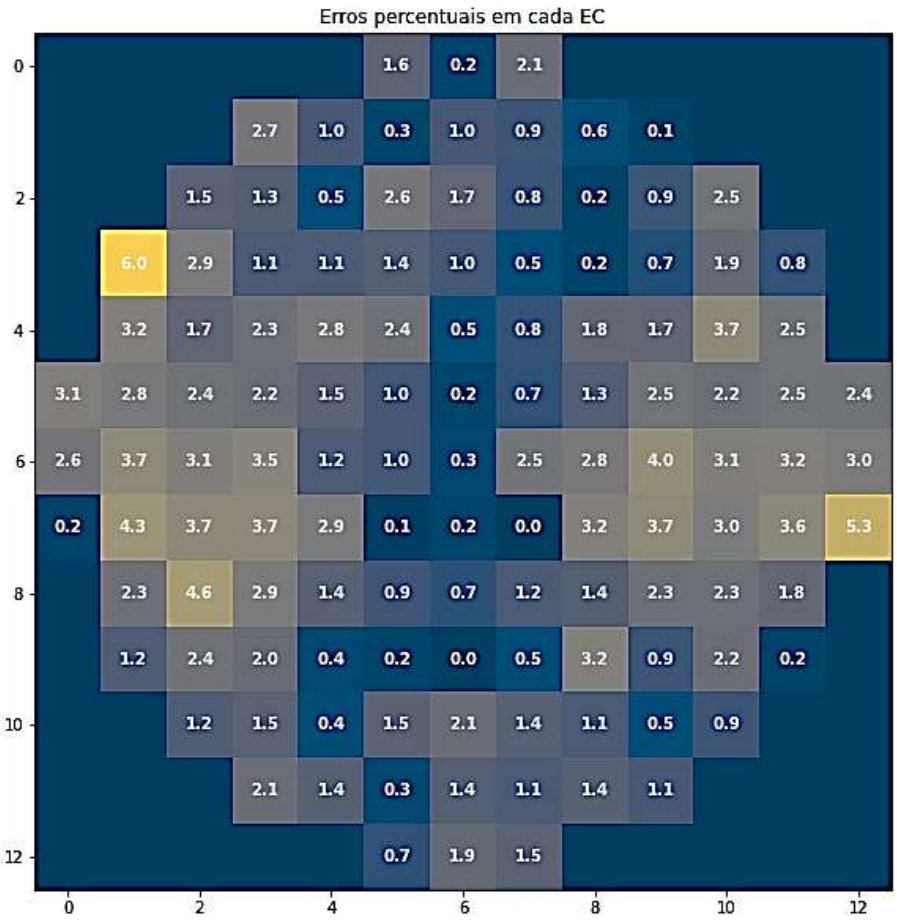

Figura 5.22: Erros percentuais para cada EC no instante de queima 3,99 dias com função de ativação tangente hiperbólica.

Fonte: Elaboração própria.

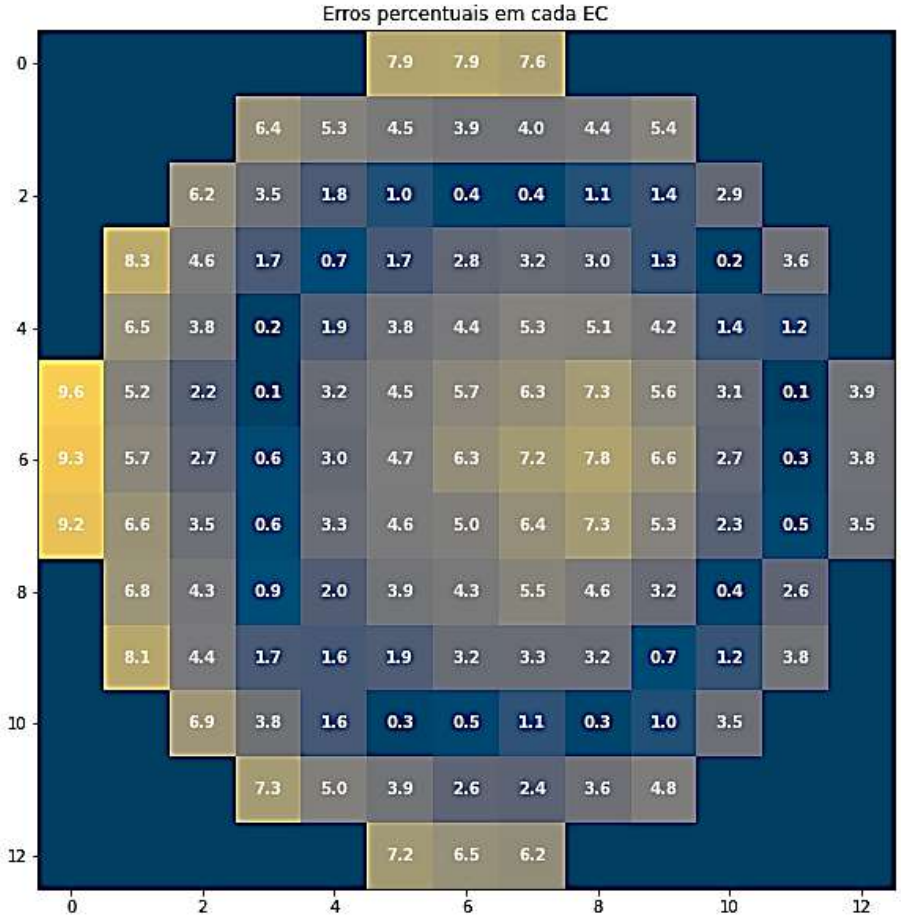

Figura 5.23: Erros percentuais para cada EC no instante de queima 3,99 dias com função de ativação sigmoide.

Fonte: Elaboração própria.

Utilizando o algoritmo de treinamento Adam na segunda base de dados, a RNA com 33 entradas apresenta na maior parte dos resultados, erros relativos máximos menores que a RNA com 36 entradas, porém a distribuição dos erros é melhor para a rede com 36 entradas, para melhor visualização a tabela 5.5 apresenta os erros máximos das redes com 36 e 33 entradas com a utilização do algoritmo de treinamento Adam.

Tabela 5.5: Erros relativos para segunda base de dados com algoritmo de treinamento Adam.

Fonte: Elaboração própria.

Erros relativos máximos		
36 detectores		
	Adam	
Instante de queima	Sigmoide	Tangente hiperbolica
3,99	9,5	6,4
33 detectores		
	Adam	
Instante de queima	Sigmoide	Tangente hiperbolica
3,99	9,6	6,0

Para a rede com algoritmo de treinamento SGD, os resultados dos erros obtidos são demonstrados nas Figuras 5.24 e 5.25:

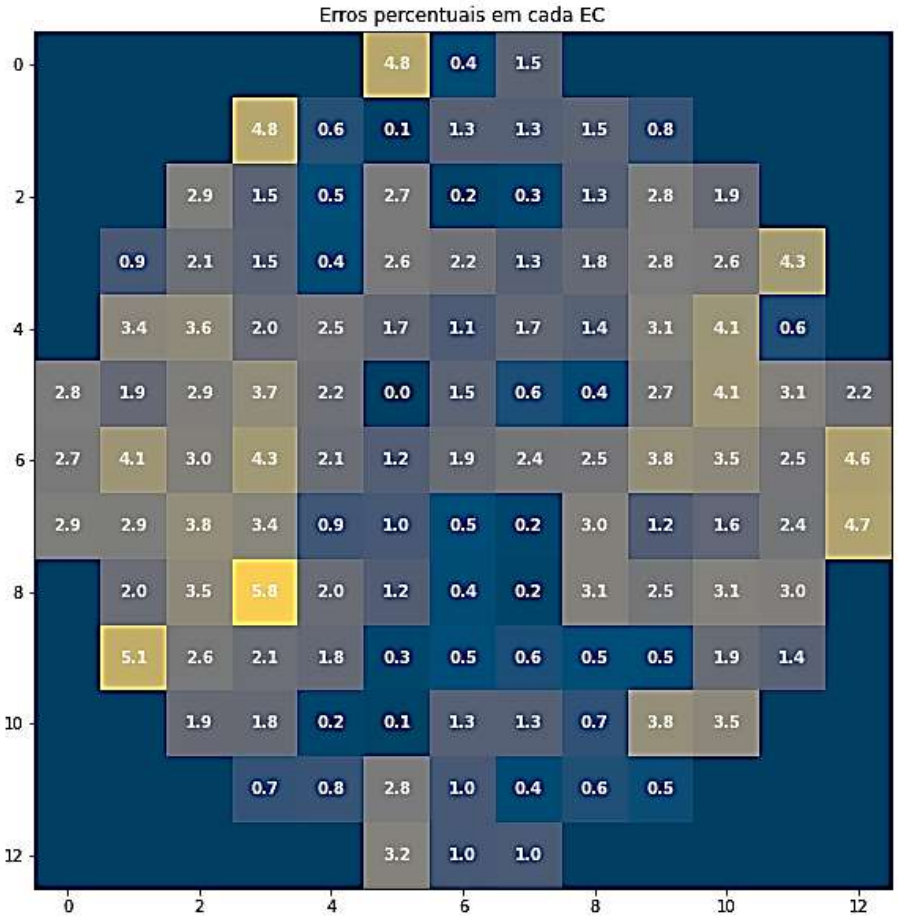

Figura 5.24: Erros percentuais para cada EC no instante de queima 3,99 dias com função de ativação tangente hiperbólica.

Fonte: Elaboração própria.

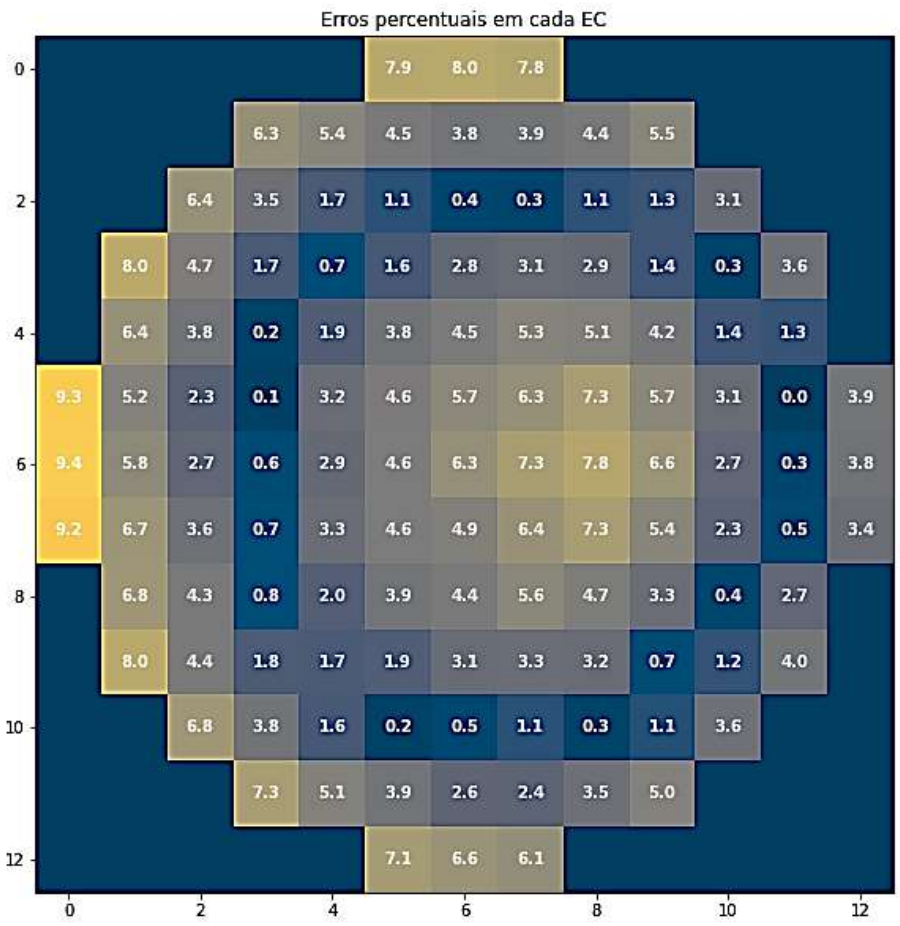

Figura 5.25: Erros percentuais para cada EC no instante de queima 3,99 dias com função de ativação sigmoide.

Fonte: Elaboração própria.

Por fim, para visualização da implementação da metodologia, a figura 5.26 demonstra valores e erros relativos das potências presentes no manual de Angra-1 e os valores previstos pela RNA, em 1/8 do núcleo do reator, referentes ao nono instante de queima (159,91 dias), onde foi utilizado o algoritmo de treinamento Adam e função de ativação tangente hiperbólica.

1,2				
1,1815				
2%				
1,17	1,2			
1,1731	1,21361			
0%	1%			
1,19	1,2	1,18		
1,2152	1,28925	1,22178		
2%	7%	4%		
1,18	1,17	1,18	1,18	
1,26826	1,19505	1,22367	1,07102	
7%	2%	4%	9%	
1,12	1,16	1,04	0,99	0,65
1,14113	1,15647	1,02026	0,99288	0,615913
2%	0%	2%	0%	5%
1,02	1	0,86	0,59	
1,00149	0,930353	0,791804	0,530452	
2%	7%	8%	10%	
0,7	0,57			
0,6426	0,516278			
8%	9%			

Figura 5.26: Valores de potência do manual e previstos pela rede com respectivos erros relativos percentuais no intervalo de queima de 159,91 dias.
Fonte: Elaboração própria.

Os valores de distribuição das densidades de potência fornecidas pela RNA e os valores presentes no manual de operação do ciclo 1 de Angra-1 no geral possui erros relativos percentuais maiores que os erros relativos dos valores simulados pelo SERPENT e os valores do manual. Tal comportamento pode ser explicado pelo fato de que a rede é treinada com os resultados simulados. Assim os valores por ela computados, não são idênticos aos valores simulados, consequentemente os erros entre esses valores e os valores do manual serão maiores.

6. Conclusões

O trabalho desenvolvido nesta dissertação visa o desenvolvimento de metodologias que auxiliem no processo de monitoramento em tempo real das densidades de potência radial em reatores do tipo PWR's contribuindo na ampliação da margem de manobras operacionais destes tipos de reatores. Para tal objetivo foi desenvolvido um método que permita obter a distribuição de potência radial do núcleo do reator, por meio de respostas simuladas dos detectores internos.

Os resultados dos detectores internos, referentes ao SDNI, foram obtidos por meio de simulações, referentes a um reator com configurações similares as do ciclo 1 do reator de Angra-1, quais foram realizadas para 18 instantes de queima, abrangendo 346 dias de operação do reator em plena potência.

Em seguida foram realizados treinamentos de redes neurais do tipo perceptron de múltiplas camadas para obtenção da melhor configuração da rede a ser implementada. Onde foi definido como parâmetro de avaliação da rede um erro relativo percentual máximo de 10%.

Após os treinamentos, onde foram mantidas as mesmas bases de dados, é perceptível o melhor desempenho da RNA treinada com maior número de entradas, como previamente mencionada a rede encontra mais facilidade para realizar os ajustes dos seus parâmetros e encontrar melhores resultados com uma maior quantidade de dados. Em relação a arquitetura da rede, foram alterados apenas dois parâmetros, os algoritmos de treinamento e as funções de ativação dos neurônios artificiais das camadas ocultas. Ao analisar os resultados, é possível saber qual arquitetura de rede neural permitiu melhor adaptação da rede aos dados.

Dessa forma, três das quatros estruturas de RNA apresentou valores razoáveis de erros entre os valores desejados e o valor obtido pela rede, contudo, a estrutura de rede que dispõe do algoritmo de treinamento SGD e da função de ativação sigmoide apresentou valores muito elevados de erros.

Apesar do algoritmo de treinamento SGD apresentar bons resultados nos treinamentos que foi implementado em conjunto com a função de ativação tangente hiperbólica, onde os valores de erros mostraram-se razoáveis, ao ser utilizado em conjunto com a função de ativação sigmoide, apresentou erros máximos próximos a 30%, o que é inviável para a implementação da metodologia.

A estrutura de rede que utiliza do algoritmo de treinamento Adam e da função de ativação sigmoide, apresenta melhores valores de erros para a primeira base de dados, porém para a segunda base de dados os valores computados pela rede não foram tão satisfatórios, onde erros próximos a 10% foram apresentados. Dessa forma, a melhor configuração de rede neural para a implementação da metodologia, foi a RNA que contém o algoritmo de treinamento Adam e as funções de ativação tangente hiperbólica para os neurônios das camadas ocultas. Visto que apresentou melhor distribuição de erro para ambas bases de dados.

Em relação ao desempenho da rede neural artificial, é possível uma melhora dos resultados ao testar outros parâmetros como outros algoritmos de treinamento, uma intercalação entre as funções de ativação nas camadas ocultas ou alterações na taxa de aprendizagem da rede, com intuito de viabilizar erros menores ou iguais a 5%.

Assim, os resultados encontrados neste trabalho nos permitem concluir que a implementação da metodologia do uso de redes neurais artificiais para a previsão e reconstrução da distribuição de potência radial a partir de medidas realizadas pelos detectores internos é viável. Uma vez que, com uma base de dados suficientemente robusta, a rede consegue obter resultados de potências aceitáveis de instantes de queima que não foram apresentados a ela.

Dessa forma, os resultados indicam ser possível a implementação da metodologia para outros ciclos de um reator do tipo PWR, pois, uma vez que se possua as características de operação de um determinado reator, é possível adquirir as respostas dos detectores internos através de códigos de simulações, como por exemplo o código de Monte Carlo SERPENT. Assim realizar o treinamento da RNA, e, por fim, reconstruir a distribuição das potências, sendo possível a obtenção de tais valores com erros razoáveis com a utilização de uma menor quantidade de detectores internos, uma vez que a rede que utiliza das 33 respostas dos detectores internos como entradas, ao empregar o algoritmo de treinamento Adam na sua configuração, apresentou resultados satisfatórios, pois os erros percentuais apresentados pela rede, não ultrapassaram os 10%.

E, como prospecção do trabalho, temos a implementação de uma rede neural artificial de forma *online* onde a previsão e reconstrução da distribuição das densidades de potência serão realizadas através de dados em tempo real provenientes dos detectores internos do núcleo de um reator do tipo PWR. Além disso, mostra-se relevante o estudo de viabilidade da rede com a utilização de uma menor quantidade de dados de entrada

fornecidos, ou seja, utilizar de um menor número de respostas dos detectores internos para a reconstrução da distribuição das potências.

7. Referências

BARBON JR, S. Fundamentos de Inteligência Artificial [5COP099]. Londrina: UEL, s.d.

BINIELI, M., *Machine Learning: An Introduction to Mean Squared Error and Regression Lines*, freeCodeCamp, 2018. Disponível em:< https://www.freecodecamp.org/news/machine-learning-mean-squared-error-regression-line-c7dde9a26b93/>.

BRAGA, A.P., CARVALHO, A.P.L.F., LUDEMIR, T.B. *Redes Neurais Artificiais: Teoria e Aplicações.* 1 ed. Rio de Janeiro, LTC – Livros Técnicos e Científicos Editora S.A, 2000.

BROWNLEE, J. *Deep Learning With Python*. v. 1.7, 2016.

CÁCERES, S. Representação e Modelagem. São Paulo: UNIP, s.d.

CASTANHO, C. T. Regressão Linear. São Paulo: UNIFESP, 2017.

CECCON, D., *Funções de Ativação: Definição, Características, e quando usar cada uma.* Ia Expert Academy, 2020. Disponível em:< https://iaexpert.academy/2020/05/25/funcoes-de-ativacao-definicao-caracteristicas-e-quando-usar-cada-uma/>.

CHEIN, F. Introdução aos modelos de regressão linear. Brasília: ENAP, Coleção Metodologias de Pesquisa, 2019.

DIDÁTICA TECH. *Redes Neurais em Problemas de Regressão*. s.d. Disponível em:< https://didatica.tech/redes-neurais-em-problemas-de-regressao/>.

DUDERSTADT, J.J., HAMILTON, L.J. *Nuclear Reactor Analysis*. 1 ed. New Jersey, John Wiley & Sons, Inc, 1976.

FERNEDA, E. Redes Neurais e sua Aplicação em Sistemas de Recuperação de Informação. *Ciência da Informação*, v.35, n. 1 (Jan/Abr), pp. 25-30, 2006.

FILHO, M.L.O. A Utilização da Regressão Linear Como Ferramenta Estratégica Para a Projeção dos Custos Produção. In: *IX Congresso Brasileiro de CustosI*, São Paulo, SP, Out. 2002.

FLECK, L. *et al*. "*Redes Neurais Artificiais: Princípios Básicos*. *Revista Eletrônica Científica Inovação e Tecnologia*, v. 1, n. 13(Jan/Jun), pp. 47-57. 2016.

FLEMING, W. H.; YOUNG, L. C. *Generalized surfaces with prescribed elementar boundary*. Madinson: University of Wisconsin, 1956.

FONSECA, R.W., DIDONÉ, E.L., PEREIRA, F.O.R. *Modelos de Predição da Redução do Consumo Energético em Edifícios que Utilizam a Iluminação Natural Através de*

Regressão Linear Multivariada e Redes Neurais Artificiais. *Ambiente Construído*, v. 12, n. 1 (Jan/Mar), pp. 163-175, 2012.

FONTANA, E. *Introdução aos Algoritmos de Aprendizagem Supervisionada*. Paraná: Universidade Federal do Paraná, 2020.

FREITAS, A.C. *Estudo do Processo de Densificação de Combustíveis Urânio-Érbio para Reatores do Tipo Água Leve.* Tese de M.Sc., IPEN/USP, São Paulo, SP, Brasil, 2017.

GATTAS, M. *DNN- Uma introdução ao Algoritmo de gradientes descendente*. 2022. Disponível em: https://www.youtube.com/watch?v=aQIgDzDWvR0. Acesso em: 25/10/2022.

_____. *DNN – Uma Introdução ao Algoritmo de Gradiente Descendente*. 2021. Disponível em:< https://www.youtube.com/watch?v=VrDiMsVyLMw>. Acesso em: 24/10/2022.

GEEKSFORGEEKS. *Linear Regression (Python Implementation).* 2022. Disponível em: < https://www.geeksforgeeks.org/linear-regression-python-implementation/?ref=leftbar-rightbar>.

GODFREY, A.T. *VERA Core Physics Benchmark Progression Problem Specifications.* Oak Ridge National Laboraty, 2014.

GONÇALVES, A. C. *Física de reatores nucleares*. 2021.

GRANATYR, J. *Redes neurais multicamada*. EXPERT, s.d.

_____. *Tipos de Aprendizagem de Máquina*. EXPERT, s.d.

HAYKIN, S. *Redes Neurais, Princípios e Prática.* 2 ed. São Paulo, Artmec, 2001.

_____. *Neural Networks and Learning Machines.* 3 ed. New Jersey, Pearson Education, Inc, 2009.

HORNIK, K.; WHITE, M. S.; STINCHCOMBE, M. *Multilayer Feedfoward Networks are Universal Approximators.* Estados Unidos: v. 2, p. 359-366, 1989.

INTERNATIONAL ATOMIC ENERGY AGENCY. Nuclear power reactors in the world. Vienna: n. 2, 2021.

KERAS. Keras. S.d. Disponível em https://keras.io/ Acesso: novembro de 2022.

KINGMA, D. P.; LEI BA, J. *Adam: A method for stochastic optimization.* ICLR: v. 9, jan. 2017.

LAMARSH, J.R. *Introduction to Nuclear Reactor Theory*. 1 ed. Boston, Addison-Wesley Publishing Company, 1972

_____. *Introduction to Nuclear Engineering.* 3 ed. New Jersey, Prentice Hall, Inc, 2001.

LEPPÄNEN, J. *Development of a New Monte Carlo Reactor Physics Code*. VTT Publications 640, 228 p., ESPOO, 2007.

LI, Z. *et al. Development and validation of a PWR on-line power-distribution monitoring system NECP-ONION*. Nuclear Engineering and Design 322, p. 104-115, 2017.

LISH, K.C. *Nuclear Power Plant Systems and Equipment*. 2 ed. New York, Industrial Press Inc., 1972.

LIU, Y. L. A *et al. Uso de Rede Neural Percéptron Multi-Camadas na Classificação de Patologias Cardíacas*. Sociedade Brasileira de Matemática Aplicada e Computacional, v.9, n. 2, p. 255-264, 2008.

MACHADO, I. F.; DRIEMEIER, L. PMR5251 – *Avaliação do Comportamento Mecânico de Materiais Utilizando uma Abordagem de ML*. São Paulo: POLI USP, 2020.

MAKAI, M., VÉGH, J. *Reactor Core Monitoring*. 1 ed. Cham, Springer International Publishing, 2017.

MASTERSON, R.E. *Introduction to Nuclear Reactor Physics*. 1 ed. Florida, Taylor & Francis Group, LLC, 2018.

MEDEIROS, F.S.B., BIANCHI, R.C. A Aplicação do Método Regressão Linear Simples na Demanda de Produtos Sazonais: Um Estudo de Caso. *Disciplinarum Scientia.* Série: Ciências Sociais Aplicadas, S. Maria, v.5, n. 1, pp. 35-53, 2009.

MICKUS, I. *Towards Efficient Monte Carlo Calculations in Reactor Physics*. Suécia: KTH Royal Institute of Technology, 2021.

MIGUEZ, G.A. *Otimização do Algoritmo de Backpropagation Pelo Uso da Função de Ativação Bi-Hiperbólica.* Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2012.

MOREIRA, C. SINAPSE. *Revista de Ciência Elementar*, v. 2, n. 4 (Dezem), pp. 317 – 320, 2014.

OLIVEIRA, L.F. *Transmissão Sináptica. Revista Brasileira de Anestesiologia*, v. 44, n. 1(Jan/Fev), pp. 25-33, 1994.

OLIVEIRA, S.S.B *et al.* Energia Nuclear: Vantagens e Desvantagens. In: Simpósio de Engenharia de Produção de Sergipe. São Cristovão, SE, Set. 2016.

O'REILLY. Artificial Intelligence By Example by Denis Rothman. 2023. Disponível em https://www.oreilly.com/library/view/artificial-intelligence-by/9781788990547/97eeab76-9e0e-4f41-87dc-03a65c3efec3.xhtml Acesso em: janeiro de 2023.

PATERNELLI, L. A. *Regressão linear e correlação*. Cap. 9, p. 1 – 11, s.d.

PAULI, S.T.Z., KLEINA, M., BONAT, W.H. Técnicas de Redes Neurais Artificiais e Regressão Linear Múltipla Aplicadas na Previsão de Valores de Importação e Exportação no Setor de Bebidas no Brasil. In: *IX Congresso Brasileiro de Engenharia de Produção*, Ponta Grossa, PR, Dezem. 2019. Disponível em: < http://aprepro.org.br/conbrepro/2019/anais/>.

PUC-Rio. *Redes Neurais Artificiais*. Puc-Rio, Certificação Digital Nº 0610775/CA.

REIS, B., *Redes Neurais – Funções de Ativação*, Laboratório Imobilis – Computação Móvel, 2016. Disponível em:< http://www2.decom.ufop.br/imobilis/redes-neurais-funcoes-de-ativacao/>.

RODRIGUES, S. C. A. *Modelo de Regressão Linear e suas Aplicações*. Covilhã: Universidade da Beira Interior, 2012.

SADDE, M.S. *Determinação da Resposta nos Detetores Externos de um Reator PWR Para Diferentes Configurações de Queda de Barras de Controle*. Tese de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2000.

SAEED, A., RASHID, A. *Development of Core Monotoring System for a Nuclear Power Planting Using Artificial Neural Network Technique*. *Elsevier – Annals of Nuclear Energy*. v. 144, (Maio), 2020.

SEFIK ILKIN SERENGIL. *Hyperbolic Tangent as Neural Network Activation Function.* 2017. Disponível em https://sefiks.com/2017/01/29/hyperbolic-tangent-as-neural-network-activation-function/ Acesso: dezembro de 2022.

SILVA, C.A.P.S., ALMEIDA, D.F.S., *Métodos dos Mínimos Quadrados e Regressão Linear com Ambiente Jupyer e Linguagem Python*. In: Semana Acadêmica, Encontro regional de Educação Matemática e Encontro de pós-graduação lato-senso em Educação e Educação Matemática. Araguaína, TO, 2019.

SOMERVILLE, J.J., DEMAS, N.G. *The Nuclear Designo of The Angra Nuclear Power Plant, Cycle 1*. 1 ed. Pennsylvania, Westinghouse Eletric Corporation, 1976.

SOUZA, R. M. G. P.; MOREIRA, J. M. L; *Neural network correlation for power peak factor estimation.* Annals of Nuclear Energy, v. 33, p. 594-608, 2006.

SOUZA, T. J.; MEDEIROS, J. A. C. C.; GONÇALVES, A. C. *Identification model of an accidental drop of a control rod in PWR reactors using thermocouple readings and radial basis function neural networks.* Annals of Nuclear Energy, 103, p. 204-211, 2017.

SPÖRL, C., CASTRO, E.G., LUCHIARI, A. *Aplicação de Redes Neurais Artificiais na Construção de Modelos de Fragilidade Ambiental*. Revista do Departamento de Geografia – USP, v.21, (Ago), pp. 113-135, 2011.

STACEY, W.M. *Nuclear Reactor Physics.* 2 ed. Weinheim, Wiley-Vch Verlag GmbH & Co., 2007.

TECNOBLOG. O que é uma API? [Guia para iniciantes]: Saiba o que é uma API, como funciona e as principais vantagens do conjunto de padrões que é fundamental para a web moderna. 2021. Disponível em https://tecnoblog.net/responde/o-que-e-uma-api-guia-para-iniciantes/ Acesso: novembro de 2022.

TENSORFLOW. Criar modelos de machine learning no nível de produção com o TensorFlow. S.d. Disponível em https://www.tensorflow.org/?hl=pt-br Acesso: dezembro de 2022.

TODREAS, N.E., KAZIMI, M.S. *Nuclear Systems*. 2 ed. Florida, Taylor & Francis Group, LLC, 2011.

VAZ, A. L. *Gradientes Descendentes na prática*. 2022. Disponível em:<https://medium.com/data-hackers/gradientes-descendentes-na-pr%C3%A1tica-melhor-jeito-de-entender-740ef4ff6c43#:~:text=A%20gradiente%20descedente%20%C3%A9%20um,os%20par%C3%A2metros%20do%20nosso%20modelo.>. Acesso em: 28/11/2022.

YORIYAZ, H. Método Monte Carlo: *Princípios e Aplicações em Física Médica*, Revista Brasileira de Física Médica, v. 3, n. 1, pp. 141-149, 2009.

ZHANG, A. *et al. Dive into deep learning.* Livro interativo, s.d.. Disponível em:<https://d2l.ai/>

APÊNDICE A: Cálculos das densidades atômicas para implementação no código SERPENT

Utilizando das porcentagens de massa fornecidas na tabela 3.1 do manual de operação do ciclo 1 de Angra-1, foram calculadas densidades atômicas dos nuclideos que compõem a pastilha de UO_2. A partir da equação presente na dissertação do pesquisador Artur Cesar de Freitas (2017), foram calculadas as densidades teóricas da pastilha.

$$Dt = \frac{\Sigma(nA)}{V * N_A} \qquad (A.1)$$

Onde:

Σ(nA): Somatória das massas molares;

V: Volume da célula unitária

N_A: Número de Avogadro

Em seu trabalho, Freitas (2017) esclarece que a pastilha do UO_2 possui uma estrutura cristalina do tipo fluorita, tipo CaF_2, onde os átomos de urânio formam uma estrutura cúbica de face centrada e o oxigênio forma uma estrutura cúbica. A figura A.1 apresenta representação da pastilha, onde as bolas de cor azul representam os átomos de urânio e as de cor vermelha os átomos de oxigênio.

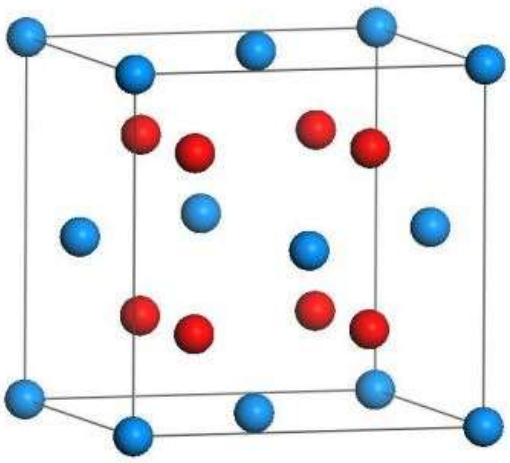

Figura A.1: Estrutura cristalina de UO_2.

Fonte: Freitas (2017).

Assim Freitas (2017) esclarece que ao analisar a estrutura cristalina, é possível deduzir que um quarto da diagonal principal do cubo formado pela estrutura cristalina é

composta pelo somatório do raio do oxigênio com o do urânio, portanto as seguintes equações são apresentadas:

$$\frac{1}{4}\, d = \frac{a\sqrt{3}}{4} \qquad \text{(A.2)}$$

$$r_{urânio} + r_{oxigênio} = \frac{a\sqrt{3}}{4} \qquad \text{(A.3)}$$

$$a = \frac{4}{\sqrt{3}}(r_{urânio} + r_{oxigênio}) \qquad \text{(A.4)}$$

Freitas (2017) disponibiliza o valor dos raios iônicos do urânio e do oxigênio, que são respectivamente 0,10011 nm e 0,13675 nm, ao substituir os valores na equação encontramos o valor de "a" igual a 0,5470 nm, tal valor permite encontrar o valor do volume da célula unitária.

$$V = 0{,}163667\ nm^3$$

Realizando a transformação para cm^3 temos:

$$V = 1{,}63667 * 10^{-22}$$

Substituindo os valores na equação A.1, a densidade teórica é:

$$Dt = \frac{(M_U * 4 + 8 * 15{,}999)}{(1{,}63667 * 10^{-22} * 0{,}6022 * 10^{24})}$$

Com isso em vista, para a obtenção do valor da densidade teórica da pastilha, é necessário encontrar a massa atômica do urânio. A respeito disso Lamarsh (1983) relata que em algumas aplicações nucleares a composição isotópica de um elemento é alterada através de meios artificiais como por exemplo, o urânio usado como combustível em reatores nucleares tem sua abundância de ^{235}U enriquecida. Nesse caso, o enriquecimento é normalmente especificado em porcentagem de massa (w/o) e, devido a tais alterações na composição isotópica do nuclídeo, é necessário calcular sua massa molar a partir das abundancias de cada um dos seus isótopos.

Portanto para tais cálculos, é pertinente registrar a equação A.5 apresentada por Lamarsh (1983):

$$\frac{1}{M} = \frac{1}{100} * \Sigma \frac{w_i}{M_i} \qquad \text{(A.5)}$$

Onde:

M: Massa atômica do elemento;

w_i: Frações de massa dos isótopos no elemento;

M_i: Massa atômica dos isótopos.

Dessa forma, a fim de exemplificar o método para calcular a massa atômica do urânio no composto de UO_2 é utilizado um enriquecimento de 2,1 % (w/o):

$$\frac{1}{M} = \frac{1}{100} * \left(\frac{2{,}1}{235{,}0409} + \frac{97{,}9}{238{,}0507}\right)$$

$$M = 237{,}9867$$

Logo com o valor da M encontrado, é possível obter o valor da densidade teórica da pastilha combustível para 2,1 w/o.

$$Dt = \frac{(237{,}9867 * 4 + 8 * 15{,}999)}{(1{,}63667 * 10^{-22} * 0{,}6022 * 10^{24})}$$

$$Dt = 10{,}95714 \ {}^{g}/_{cm^3}$$

Mediante ao apresentado é necessário calcular a densidade teórica da pastilha combustível com diferentes enriquecimentos, uma vez que Lamarsh (1983), especifica que com qualquer mudança de enriquecimento o valor encontrada pela equação A.5 é alterado.

A vista disso, para calcular as densidades atômicas dos nuclídeos para inserção no código SERPENT, é necessário empregar a seguinte equação apontada por Lamarsh (1983):

$$N = \frac{\rho N_A}{M} \qquad \text{(A.6)}$$

Onde:

N: Densidade atômica;

ρ: Densidade de massa;

N_A: Número de Avogrado;

M: Massa atômica.

Dessa forma, para o cálculo da densidade atômica do U_{235} com 2,1 % de enriquecimento no composto, Lamarsh (1983) aponta que primeiramente é necessário a obtenção da densidade da pastilha combustível, para posteriormente obter a fração de massa de urânio no composto:

$$M_U + M_O = (237{,}9867 + 2 * 15{,}999)$$

$$M_T = 269{,}9848$$

Fração de massa do urânio:

$$,\frac{M_U}{M_T} = \frac{237{,}9867}{269{,}9848} = 0{,}881481 = 88{,}1481\%$$

Portanto a densidade de massa de urânio no composto da pastilha é:

$$D_{UO_2} * 0{,}881481 = 9{,}175585\,{}^{g}/_{cm^3}$$

Dessa forma é possível encontrar a densidade do ^{235}U e ^{238}U da seguinte forma:

$$D_{U_{235}} = 2{,}1\% * 9{,}175585\,{}^{g}/_{cm^3} = 0{,}021 * 9{,}175585 = 0{,}192687\,{}^{g}/_{cm^3}$$

$$D_{U_{238}} = 97{,}9\% * 9{,}175585\,{}^{g}/_{cm^3} = 0{,}979 * 9{,}175585 = 8{,}98289\,{}^{g}/_{cm^3}$$

Por fim é empregada a equação A.6 para calcular a densidade atômica do ^{235}U e ^{238}U, sendo:

$$N_{U_{235}} = \frac{0{,}192687 * 0{,}6022 * 10^{24}}{235{,}0439}$$

$$N_{U_{235}} = 0{,}0493602 * 10^{22}\,{}^{átomos}/_{cm^3}$$

Foram realizados os mesmos cálculos para encontrar as densidades dos isótopos que compõem a pastilha combustível sendo eles: ^{234}U, ^{235}U, ^{236}U e ^{238}U e oxigênio.

Neste trabalho para obtenção de simulações mais precisas possíveis, os cálculos descritos acima foram realizados para cada região de enriquecimento.