

UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITECNICA

CELESTINO PAULO TCHIPONGUE KALUEYO

Redes neurais artificiais para localização de faltas em linhas de transmissão
com derivação

Versão corrigida

São Paulo

2023

CELESTINO PAULO TCHIPONGUE KALUEYO

Redes neurais artificiais para localização de faltas em linhas de transmissão
com derivação

Versão corrigida

Dissertação apresentada à Escola Politécnica
Universidade de São Paulo para obtenção
do título de Mestre em ciências.

Área de Concentração: Sistema de Potência

Orientador: Prof. Dr. Silvio Giuseppe Di Santo

São Paulo

2023

Ficha Catalográfica

KALUEYO, Celestino Paulo Tchipongue

Redes neurais artificiais para localização de faltas em linhas de transmissão com derivação, São Paulo / Celestino Paulo Tchipongue Kalueyo; orientador, Silvio Giuseppe Di Santo. São Paulo : Escola Politécnica, 2023
116 p.

Dissertação (mestrado em Engenharia Elétrica) - Programa de Pós-graduação Engenharia Elétrica, escola Politécnica, departamento de Engenharia Elétrica, Universidade de São Paulo, São Paulo, 2023


1. Redes neurais artificiais 2. Linhas de transmissão de energia elétrica
3. Sistemas elétricos de potência 3. Proteção de sistemas elétricos I.
Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Energia e Automação Elétricas II. t.

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 30 de maio de 2023

Assinatura do autor: Celestino Kalueyo

Assinatura do orientador: 

Catálogo-na-publicação

Kalueyo, Celestino Paulo Tchipongue
Redes neurais artificiais para localização de faltas em linhas de transmissão com derivação / C. P. T. Kalueyo -- versão corr. -- São Paulo, 2023.
116 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Energia e Automação Elétricas.

1.Redes neurais 2.Linhas de transmissão 3.Sistemas de proteção elétricas I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Energia e Automação Elétricas II.t.

KALUEYO, Celestino Paulo Tchipongue

Redes neurais artificiais para localização de faltas em linhas de transmissão com derivação

Dissertação apresentada à Escola Politecnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia Elétrica.

Aprovada em:

Banca Examinadora

Prof. Dr. _____

Instituição: _____

Julgamento: _____

Prof. Dr. _____

Instituição: _____

Julgamento: _____

Prof. Dr. _____

Instituição: _____

Julgamento: _____

Dedico esse trabalho para:

O meu pai pelo incentivo e o apoio financeiro para a concretização dos meus objetivos académicos, a minha mãe, ao meu tio Maurício e a minha tia Adriana, aos meus irmãos Luciano Kalueyo e Honorato Tenente.

AGRADECIMENTOS

Ao meu orientador Prof. Dr. Silvio Giuseppe Di Santo pela confiança, orientação e os conhecimentos transmitidos ao longo de toda a formação.

Aos professores Drs. Carlos Eduardo de Moraes Pereira e Marcos Rosa Dos Santos pela presença e principalmente pelas sugestões na etapa da qualificação.

À CAPES pelo auxílio financeiro.

Aos meus amigos Paulo Venâncio, Edson Tchivela, Generoso Gouveia, Patrese Xavier e Norton Liz pelo apoio moral.

Não é sempre que a maioria está certa

RESUMO

Este trabalho descreve um sistema de localização de faltas em linhas de transmissão de energia elétrica que possuem derivações. O sistema utiliza modelos de inteligência artificial, mais especificamente redes neurais artificiais. Os sinais de tensão e corrente provenientes de dois terminais, um local e outro remoto, são fornecidos como entrada para o modelo de rede neural.

Além da localização de faltas, o trabalho também aborda a detecção de faltas e a classificação do tipo de falta. Isso significa que o sistema não apenas identifica a presença de um curto-circuito na linha de transmissão, mas também é capaz de determinar o tipo específico de falta.

Para implementar a técnica, foi realizada uma simulação da linha de transmissão no software ATP (Alternative Transients Program) e desenvolvido um algoritmo no MATLAB para automatizar o processo de simulação. Esse algoritmo gerou dados correspondentes a casos com presença de falta (dados faltosos) e casos sem falta (dados não faltosos) no sistema.

Além disso, outro algoritmo foi desenvolvido no MATLAB com o objetivo de criar um banco de dados e armazenar os dados obtidos a partir das simulações. Esse banco de dados armazena informações como as características das faltas, parâmetros do sistema, resultados das medições, entre outros dados relevantes para a análise e o desenvolvimento da técnica de localização de faltas.

Os dados gerados foram utilizados para alimentar os modelos de redes neurais artificiais. Esses modelos possuem 24 entradas, que representam os valores fasoriais das tensões e correntes da linha de transmissão. Essas entradas são processadas pelos neurônios do modelo, que realizam cálculos e tomam decisões com base nos padrões dos dados.

Quanto à saída dos modelos, o modelo de localização e o modelo de detecção possuem apenas um neurônio na camada de saída. Esse neurônio é responsável por fornecer uma resposta binária indicando a presença ou ausência de falta e como o local da ocorrência de falta na linha de transmissão.

Já o modelo de classificação possui 4 neurônios na camada de saída. Cada neurônio representa um tipo específico de falta, como curto-circuito fase-terra, curto-circuito fase-fase, entre outros. Cada neurônio irá produzir uma saída correspondente à probabilidade de a falta pertencer àquela categoria.

O que caracteriza especificamente essa técnica é o fato de que, no processo de localização de faltas, são utilizados apenas dados correspondentes ao trecho específico que está sendo analisado, levando em consideração seus referenciais e características individuais. Isso significa que a técnica se concentra nos dados relevantes ao trecho em questão, evitando a inclusão de informações desnecessárias de outros trechos ou referências. Isso permite uma análise mais precisa e focada na localização da falta dentro do trecho específico da linha de transmissão.

Palavras-chave: Redes neurais. Linhas de transmissão. Sistemas de proteção elétricas.

ABSTRACT

This work describes a fault location system in power transmission lines that have taps. The system uses artificial intelligence models, more specifically artificial neural networks. Voltage and current signals from two terminals, one local and one remote, are provided as input to the neural network model.

In addition to fault location, the work also addresses fault detection and fault type classification. This means that the system not only identifies the presence of a short circuit on the transmission line, but is also able to determine the specific type of fault.

To implement the technique, a simulation of the transmission line was performed in the ATP (Alternative Transients Program) software and an algorithm was developed in MATLAB to automate the simulation process. This algorithm generated data corresponding to cases with missing (missing data) and cases without missing (non-missing data) in the system.

Furthermore, another algorithm was developed in MATLAB with the objective of creating a database and storing the data obtained from the simulations. This database stores information such as fault characteristics, system parameters, measurement results, among other relevant data for the analysis and development of the fault location technique.

The generated data were used to feed the models of artificial neural networks. These models have 24 inputs, which represent the phasor values of voltages and currents in the transmission line. These inputs are processed by model neurons, which perform calculations and make decisions based on patterns in the data.

As for the output of the models, the location model and the detection model have only one neuron in the output layer. This neuron is responsible for providing a binary response indicating the presence or absence of a fault and the location of the fault occurrence on the transmission line.

The classification model has 4 neurons in the output layer. Each neuron represents a specific type of fault, such as phase-to-ground short circuit, phase-

to-phase short circuit, among others. Each neuron will produce an output corresponding to the probability that the fault belongs to that category.

What specifically characterizes this technique is the fact that, in the fault location process, only data corresponding to the specific section being analyzed are used, taking into account its references and individual characteristics. This means that the technique focuses on data relevant to the passage in question, avoiding the inclusion of unnecessary information from other passages or references. This allows for a more accurate and focused analysis on the location of the fault within the specific stretch of the transmission line.

Keywords: Neural networks. Transmission lines. Electrical protection systems.

LISTAS DE FIGURAS

Figura 2.1 - Neurônio Biológico	14
Figura 2.2 - Córtex Humano	15
Figura 2.3 - Neuronio Artificial	16
Figura 2.4 - Função Linear	18
Figura 2.5 - Função Limiar	19
Figura 2.6 - Função Sigmóide	19
Figura 2.7 - Função tangente Hiperbolica	20
Figura 2.8 - Função de ativação <i>Relu</i>	21
Figura 2.9 - Camada Direta	22
Figura 2.10 - Multiplas Camadas	23
Figura 2.11 - Rede Recorrente	24
Figura 2.12 - Aprendizado Supervisionado Adaptado (Da Silva).....	26
Figura 2.13 - Blocos de aprendizado não supervisionado	27
Figura 2.14 - Aprendizagem por Reforço	28
Figura 2.15 - Classificação de Padrões	29
Figura 2.16 - Função de Aproximação	30
Figura 2.17 - Sistema de Controle	31
Figura 2.18 - Rede perceptron (Da Silva)	32
Figura 2.19 - Perceptron Multicamadas adaptado(Da Silva)	36
Figura 2.20 - Perceptron Multicamas Adaptado(Da Silva).....	38
Figura 2.21 - Perceptron Multicamadas(Da Silva)	40

Figura 2.22 - Representação da Retropropagação(Da Silva)	41
Figura 4.1 - Linha de transmissão utilizada para simulação e obtenção dos dados(Di Santo)	51
Figura 4.2 - Módulo de detecção de faltas	56
Figura 4.3 - Rede Neural com 24-10-1 para detecção	57
Figura 4.4 - Módulo de classificação do tipo de falta	59
Figura 4.5 - Rede Neural com 24-13-13-4 para classificação de faltas	60
Figura 4.6 - Localização de faltas	63
Figura 4.7 - Rede Neural com 24-n-n-1 para localização de faltas	65
Figura 6.1 - Desempenho do módulo da detecção	68
Figura 6.2 - Distribuição dos tipo de faltas	69
Figura 6.3 - Desempenho do módulo da classificação	70
Figura 7.1 desempenho do modelo falta fase-terra 1º trecho	94
Figura 7.2 desempenho do modelo falta fase-terra 2º trecho	95
Figura 7.3 desempenho do modelo falta fase-terra 3º trecho	95
Figura 7.4 desempenho do modelo falta fase-terra 4º trecho	96
Figura 7.5 desempenho do modelo falta fase-terra 5º trecho	96
Figura 7.6 desempenho do modelo falta fase-terra 6º trecho	97
Figura 7.7 desempenho do modelo falta fase-terra 7º trecho	97
Figura 7.8 – histograma dos erros absolutos das distâncias para faltas fase-terra a 50% de cada trecho	102
Figura 7.9 desempenho do modelo falta 3fases-terra 1º trecho	100

Figura 7.10 desempenho do modelo falta 3fases-terra 2º trecho	101
Figura 7.11 desempenho do modelo falta 3fases-terra 3º trecho	101
Figura 7.12 desempenho do modelo falta 3fases-terra 4º trecho	102
Figura 7.13 desempenho do modelo falta 3fases-terra 5º trecho	102
Figura 7.14 desempenho do modelo falta 3fases-terra 6º trecho	103
Figura 7.15 desempenho do modelo falta 3fases-terra 7º trecho	103
Figura 7.16 – histograma dos erros absolutos das distâncias para faltas fase- fase-fase a 50% de cada trecho	108

LISTAS DE TABELAS

Tabela 4.1 – Tabela da verdade classificação de faltas	76
Tabela 4.2 – Erro na classificação	77
Tabela 6.1 - taxa de acertos da detecção	69
Tabela 6.2 - taxa de acertos da classificação	71
Tabela 6.3 – Informações da linha de transmissão	82
Tabela 6.4 – características dos geradores	83
Tabela 6.5 – Sequência positiva	83
Tabela 6.6 – Sequência zero	83
Tabela 7.2 – Localização de faltas variandos as distâncias fase-terra	98
Tabela 7.3 – Erros do modelo da rede neural artificial	98
Tabela 7.4 – Teste de diferentes arquiteturas de redes neurais artificiais	99
Tabela 7.5 – Localização de faltas variandos as distâncias 3fases-terra	104
Tabela 7.6 – Erros do modelo da rede neural artificial	104

LISTA DE SÍMBOLOS

E_N^L Gerador local

E_N^R Gerador remoto

K_{eq}^L Impedância equivalente por trás da barra local

K_{eq}^R Impedância equivalente por trás da barra remota

X_i Matriz de entradas

W_i Pesos

u Potencial de ativação

θ Bias

$g()$ Função de ativação

y Saída

m Primeira camada escondida

n Segunda camada escondida

D_i Saída conhecida ou valor alvo

S_i Saída predita pela rede neural artificial

n O numero total amostrado

D Saída conhecida ou valor alvo

S Saída predita pela rede neural artificial

n O numero total amostrado

$Y_j^{(1)}$ é a saída da primeira camada intermediária

$Y_j^{(2)}$ é a saída da segunda camada intermediária

$Y_j^{(3)}$ é a saída da camada de saída do modelo da rede neural

(I_j^l) é a função de ativação contida em todos os neurônios, e é o que define que tipo de resultado deverá ser reproduzido pela saída y_j^l da RNA.

Trec Trecho

SUMÁRIO

1 INTRODUÇÃO	16
1.1 OBJETIVO	16
1.2 MOTIVAÇÃO	17
1.3 REVISÃO BIBLIOGRAFICA	18
1.3.1 Técnicas baseadas em análise em alta frequência	19
1.3.2 Princípio baseado em números de terminais monitorados	20
1.3.3 Princípios baseado em inteligência computacional	21
1.4 ESTRUTURA DO TRABALHO	24
2. Redes Neurais Artificiais	25
2.1 Introdução	25
2.2 Neurônio Biológico	27
2.3 Neuônio Artificial	28
2.4 Função de Ativação	30
2.5 ARQUITETURA	33
2.5.1 Arquitetura Direta (<i>feedforward</i>)	33
2.5.2 Arquitetura Direta de Múltiplas Camadas	35
2.5.3 Arquitetura de Rede Recorrente (Realimentação)	36
2.6 APRENDIZADO DE MÁQUINA	37
2.6.1 Aprendizagem Supervisionada	37
2.6.2 Aprendizado não Supervisionado	39
2.6.3 Aprendizado por Reforço	39
2.7 Aprendizagem	41

2.7.1 Associação de Padrões -----	41
2.7.2 Reconcimento de Padrões -----	41
2.7.3 Aproximação de Funções -----	42
2.7.4 Coontrole -----	43
2.8 Perceptron de Única Camada -----	43
2.8.1 Funcionamento do Perceptron -----	45
2.8.2 Treinamento do Perceptron -----	46
2.9 Perceptron Multicamada -----	47
2.9.1 Funcioanmento de Perceptron Multicamadas -----	47
2.9.1 Treinamento do Perceptron multicamadas -----	49
2.9.2 Desempenho de Modelos de multicamadas com Retropropagação -----	59
2.9.3 Treinamento com o Método <i>Resilent Propagation</i> -----	60
3.CONCLUSÃO -----	62
4. APLICAÇÃO DAS REDES NEURAS ARTIFICIAIS PARA LOCALIZAÇÃO DE FALTAS NAS LINHAS DE TRANSMISSÃO DE ENERGIA ELÉTRICAS--	64
4.1 Sistema simulado -----	64
4.2 Detecção da falta -----	68
4.3 Classificação do tipo de faltas -----	71
4.4 Localização de faltas -----	76
5. CONCLUSÃO -----	79
6. TOPOLOGIA DA REDE NEURAL ARTIFICIAL -----	80
6.1 Modelo de Detecção da faltas -----	81
6.2 Modelo de Classificação da faltas -----	83
6.3 Modelo de Localização de faltas -----	86
6.3.1 Sistemas equivalentes dos geradores -----	87

7. APRESENTAÇÃO DOS RESULTADOS VARIANDO AS DISTÂNCIAS DAS FALTAS	88
7.1 PARÂMETROS DOS MODELOS	88
7.1.1 Falta fase – terra	92
7.1.1.1 Resultados com variação das distâncias falta fase – terra	96
7.1.2 Faltas fase – fase - fase – terra	97
7.1.2.2 Resultados com variação das distâncias falta fase-fase-fase-terra ---	102
8 CONCLUSÃO E PORPOSTAS PARA TRABALHOS FUTUROS	109
REFERÊNCIAS	111

1 INTRODUÇÃO

1.1 OBJETIVO

O uso de redes neurais artificiais na localização de faltas em linhas de transmissão com derivação tem como objetivo principal melhorar a eficiência e a confiabilidade do sistema elétrico. Essa técnica busca identificar de maneira rápida e precisa a localização exata de defeitos nas linhas de transmissão, permitindo uma intervenção mais ágil e efetiva no reparo dos problemas.

As redes neurais artificiais são capazes de aprender e reconhecer padrões complexos nos dados de entrada, o que as torna uma ferramenta poderosa na detecção de faltas em linhas de transmissão. Elas analisam os sinais de entrada, como tensão e corrente, e identificam os padrões característicos de uma falta em um determinado trecho da linha.

Ao utilizar essa técnica, é possível reduzir o tempo necessário para identificar a localização da falta, permitindo uma resposta mais rápida por parte das equipes responsáveis pela manutenção e reparo da rede elétrica. Além disso, a identificação precisa da localização da falta contribui para a minimização de prejuízos e interrupções no fornecimento de energia elétrica.

1.2 MOTIVAÇÃO

O propósito de um sistema de potência é transmitir energia elétrica para diversos pontos, para diversas finalidades e com maior qualidade possível. Contudo, tais sistemas estão constantemente expostos a eventualidades que provocam desordem no seu estado de funcionamento normal, com isso, alterando assim, suas grandezas elétricas.

Todos esses dilemas motivaram-me a realizar este trabalho, pois, a enormes debilidades quanto a localização de faltas, após a sua ocorrência, ocasionando assim interrupções no fornecimento de energia elétrica, que podem ser de curta ou longa duração.

A linha de transmissão é de longe, o equipamento de um sistema de energia elétrica, com maior probabilidade de ocorrências de defeitos, pois, ela constitui a grade porção do sistema, e está exposto a todo tipo de fragilidade, as ocorrências podem ser de variadíssimas causas principalmente através das descargas atmosféricas.

Um dos incidentes que ocasionam grandes implicações no abastecimento de energia elétrica são os curtos-circuitos, elas são as mais graves que sucedem em consequência de uma falha do isolamento entre condutores de duas ou mais fases, o contato entre uma fase e o terra, entre duas fases e o terra, na queda de um poste, nas descargas elétricas atmosféricas e assim por diante, dentre elas, a falta fase terra decorrem com maior regularidade. Assim sendo, faz-se necessário a rapidez e precisão para a localização de falta, com o intuito de reduzir, assim, o tempo para o restabelecimento da operação normal do sistema energético

Sendo assim, o uso das redes neurais artificiais para essa finalidade são permissíveis, pois uma das suas principais qualidades é em reconhecimento de padrões. Ai isso tudo, soma-se o fato de que as redes neurais artificiais tratarem com eficiência diversos tipos de problemas inclusive os não-lineares.

1.3 REVISÃO BIBLIOGRAFICA

Neste capítulo, é feito um levantamento bibliográfico de obras existentes sobre estudos envolvendo métodos de localização de faltas.

Na literatura, existem vários estudos sobre a localização de curtos-circuitos, com o objetivo de evoluir e obter maior precisão nos cálculos matemáticos, implementar softwares para melhorar os resultados e levar em consideração as complexidades dos sistemas elétricos de potência.

De acordo com Souza (2018), existem três categorias de metodologias para a localização de faltas em linhas de transmissão:

1. Com base em componentes de alta frequência, que valem-se da técnica das ondas viajantes;
2. Com base em números de terminais monitorados ;
3. Com base em inteligência computacional.

Além das técnicas mencionadas acima, existem outras classificações para a localização de faltas, como:

Técnicas de medições de dados de pré - falta, pós - falta e durante o momento da falta;

Técnicas com base em componentes de frequência fundamental.

É importante Salientar que uma única metodologia utilizada pode conter mais de uma das categorias mencionadas.

1.3.1 Técnicas baseadas em análise em alta frequência

Esta metodologia é baseada em ondas viajantes, nas quais o tempo que um sinal leva para se propagar entre o ponto de falta e as extremidades da linha é usado para calcular a distância da falta e, em seguida, determinar a sua localização. No entanto, esse método pode gerar certos problemas, como erros quando a falta ocorre próximo do barramento ou em um ângulo de incidência zero. Além disso, o tempo necessário para obter a onda receptora e calcular a distância pode levar a atrasos no processo.

Em um estudo conduzido por (Yan et al. 2022), foi proposta uma técnica de localização de faltas em linhas de transmissão com base em ondas viajantes (TW) usando dados de um único terminal. Para resolver a dificuldade de identificação precisa da primeira onda refletida, um novo método de processamento de sinal para dados de TW foi desenvolvido com base no princípio de curtose progressiva. O segmento defeituoso é identificado pela polaridade entre a onda inicial e a segunda e a onda refletida. Por fim, a distância precisa da falta é calculada com base nos princípios de localização de faltas TW de um único terminal.

Um método de localização de faltas baseado na medição de impedância de alta frequência foi proposto por (Jia et al. 2018). Esse método é adequado para sistemas de geração distribuída. Para evitar a influência dos loops de controle com tempo de resposta em cascata de vários 10ms, é utilizada uma janela retangular curta que abrange apenas transientes de falta de 6ms. O modelo de impedância de alta frequência para geradores de indução duplamente alimentados é fornecido e testado, demonstrando uma boa precisão nos resultados. As influências da resistência da falta e dos ângulos de início do defeito também são consideradas. Além disso, o método proposto pode ser aplicado em sistemas de distribuição práticos com várias conexões usando medição multiterminal. Devido à ausência de necessidade de sincronização de dados, o método possui potencial para ser utilizado na prática.

Para localizar faltas, a Transformada Rápida de Fourier (FFT) foi empregada com o objetivo de transformar os sinais de corrente do domínio do tempo para o domínio da frequência. Nesse processo, a frequência fundamental foi utilizada para determinar o local da falta, utilizando a teoria das ondas viajantes, conforme descrito por (Mamiş, Arkan, 2011).

No estudo realizado por (Silveira, Seara, Zurn, 2001), a teoria das ondas viajantes foi estudada com o auxílio da Transformada de Wavelet (TW) para desacoplar os sinais de diversas frequências.

No estudo de (Anane et al, 2021), o método proposto depende das primeiras e sucessivas diferenças entre o tempo de incidência e reflexão da onda a partir do local da falta na linha de transmissão. Para garantir uma detecção precisa das ondas de chegada, foi explorado um conjunto de sensores de campo magnético em cada torre para medir a qualidade das ondas viajantes geradas na linha de transmissão e localizar os distúrbios transitórios decorrentes de um curto-circuito.

Para ampliar a área de abrangência, os dados da tensão e da corrente foram mensurados em diversos terminais da linha de transmissão para garantir maior eficiência no processo de localização de faltas (Panigrahi et al 2017).

(H. Livani and C. Y. Evrenosoglu, 2014) apresentam uma técnica de localização de faltas que se baseia em ondas viajantes e utiliza a transformada discreta de wavelet (TDW) para extrair transientes das tensões. Essa técnica foi aplicada em uma única extremidade da linha de transmissão híbrida.

1.3.2 Princípio baseado em números de terminais monitorados

Huynh et al. (2022) propuseram uma técnica de localização de faltas em uma linha de transmissão com compensação em série de um sistema de potência, utilizando um problema de otimização. O algoritmo Cuckoo Search (CS) foi escolhido como método de otimização, utilizando medições não sincronizadas de tensões e correntes nos terminais da linha de transmissão com compensação

em série. Os resultados obtidos pelo algoritmo CS foram comparados com os resultados de outras técnicas, como o algoritmo genético (GA) e o algoritmo de otimização de enxame de partículas (PSO). A comparação revelou que os resultados de localização de faltas obtidos pelo algoritmo CS na linha de transmissão apresentam um percentual de erro aceitável e são superiores em relação às técnicas comparadas.

(Fernando et al, 2020) utilizaram a transformada discreta de Fourier para localização de faltas, a fim de extrair os sinais de fasores das componentes da frequência fundamental de um ciclo completo, utilizando apenas dados de um único terminal.

(Mohsen Tabari, Javad Sadeh, 2022) abordaram uma proposta que visa, entre outras coisas, a resolução de problemas relacionados à localização de falhas utilizando apenas os dados de um único terminal da linha de transmissão. Os dados foram obtidos por meio da estimativa dos mínimos quadrados a partir de dois ciclos dos sinais de tensão e corrente.

O método proposto baseia-se no princípio da superposição linear e no método dos componentes simétricos, utilizando técnicas de medição de fase sincronizada de dois terminais (Zhao et al, 2011).

(Hinge, Dambhare, 2016) desenvolveram um trabalho que inclui a detecção, classificação e localização de faltas no domínio da fase, com a extração do operador de sincronização a partir das medições fasoriais disponíveis nas duas extremidades da linha de transmissão.

(Di Santo, Pereira, 2012) apresentaram um método de localização de faltas para linhas de transmissão não transpostas com múltiplas derivações, a fim de determinar informações sobre a falta, como a seção de ocorrência, a distância ou seja, o ponto de ocorrência e a resistência. Esse método baseia-se nas tensões e correntes fasoriais de pré-falta e pós-falta, medidas em estado estacionário em terminais locais e remotos.

(Anees et al, 2012) mensuraram a tensão sincronizada e a corrente nos dois terminais da linha com o objetivo de localizar faltas em qualquer ponto da linha de transmissão. O local da falta é obtido usando a matriz de impedância.

1.3.3 Princípios baseado em inteligência computacional

(Zerahny et al, 2021) Uma linha de transmissão longa foi simulada para coletar dados de falta de uma das extremidades da linha, o que tornou a abordagem economicamente viável. Utilizando a Transformada Wavelet Discreta, foram extraídas as características essenciais do tipo de falta e da sua localização. Dentre os vários tipos de wavelets testados, a wavelet de Haar foi considerada a mais adequada. Os resultados obtidos foram utilizados como recursos para treinar vários modelos de machine learning para a localização de faltas na linha de transmissão, no entanto, a rede neural artificial é o modelo que apresentou melhores resultados, com 95,9% de precisão para localização.

(M. Coban e S. S. Tezcan, 2020) fizeram um estudo a fim de determinar o local exato das faltas de curto-circuito que podem ocorrer em uma linha de transmissão de 230 kV e 100 km. Eles modelaram um algoritmo de RNA usando a técnica de retropropagação padrão *feed forward*. Após processar as informações de corrente e tensão das três fases retiradas do início da linha, as mesmas foram divididas em grupos específicos de 5 frequências e usadas como entrada para o modelo de rede neural artificial

A técnica usada por (R.C S. Ana, O.E.M. Sebastião, 2012) é de localização de faltas em linhas de transmissão com circuito duplo utilizando Redes Neurais Artificiais (RNAs), a análise é baseada em dados de tensão e corrente de ambos os circuitos, e os testes utilizam os dados oriundos dos dois terminais da linha ou de apenas um dos terminais.

Em (R. Fan, T. Yin, R. Huang, J. Lian and S. Wang, 2019) apresentaram uma nova abordagem de localização de faltas para linhas de transmissão usando técnicas modernas de aprendizado profundo. Uma rede neural convolucional

mista com estrutura de memória de longo prazo (LSTM) foi treinada para prever a distância da falta, mensurando os dados de tensão e corrente das extremidades da linha. A função convolucional, as camadas de pool e a estrutura LSTM são usadas para preservar a invariância de tradução e capturar a correlação temporal dos dados de entrada da série temporal. Técnicas avançadas de aprendizado profundo, como estimativa de momento adaptativo e dropout, são usadas para treinar com eficiência a rede neural e evitar o overfitting.

(Joorabian et al, 2004) projetaram e implementaram um localizador de faltas baseado em redes neurais artificiais para linhas de transmissão de extra alta tensão (EHV). O sistema utiliza formas de onda de tensão e corrente com falhas em apenas um dos terminais da linha. As redes de função de base radial (RBF) são treinadas com dados em várias condições de falha e usadas para classificar o tipo e localizar a falha na linha de transmissão. Os resultados obtidos a partir dos testes das redes RBF com dados de falhas simuladas e dados registrados de um sistema de 400 kV mostram claramente que essa técnica é altamente robusta e precisa.

Em (Harithaa, et al 2019) propuseram um algoritmo eficiente para detectar faltas assimétricas, classificar o tipo de falta e localizar a zona de falta em linhas de transmissão utilizando Rede Neural Artificial (RNA), que pode ser implementado em relés numéricos. O sistema completo é capaz de identificar a condição de ausência de falta, as três faltas diferentes linha-terra, faltas linha-fase e faltas duplas linha-terra, e indicando a zona na qual a falta se desenvolveu. Para localizar a falta, três zonas foram reconhecidas em cada linha de transmissão. O mesmo algoritmo foi implementado em hardware.

A técnica de redes neurais artificiais (RNA) em (Mazon et al, 2000) foi usada como metodologia que permite que seja determinada a distância na qual a falta ocorre em uma linha de transmissão de dois terminais usando os componentes fundamentais de 50/60 Hz da tensão de pré-falta e da magnitudes de corrente, que são medidos nos dois terminais.

(Belagoune et al. 2021), propuseram modelos de regressão e classificação utilizando aprendizado profundo baseado em Redes Neurais Recorrentes Profundas (RNRP) para localização, classificação e previsão de localização de falhas. Esses novos modelos exploram dados transientes de ciclos completos pré e pós-falta, em que os valores dos fasores de tensão e corrente são obtidos por meio de medições em vários terminais da linha de transmissão.

No estudo de (Carvalho et al.1999), foi desenvolvido um modelo de rede neural artificial para a localização de falhas em uma linha de transmissão de 100 km de comprimento. Os sinais de tensão e corrente antes e depois da falta, extraídos de um único terminal, foram utilizados como parâmetros para o treinamento e teste do modelo, que é do tipo supervisionado.

1.4 ESTRUTURA DO TRABALHO

1º capítulo – Revisão bibliográfica.

2º capítulo – Introdução sobre redes neurais artificiais

3º Capítulo – Conclusão sobre redes neurais artificiais

4º capítulo – Aplicação de redes neurais artificiais

5º Capítulo – Conclusão sobre aplicação de redes neurais artificiais

6º Capítulo - Topologia da Rede Neural Artificial

7º Capítulo - Apresentação Dos Resultados Variando as distâncias das faltas

8º Capítulo – Conclusão da dissertação e Propostas para trabalhos futuros

2. Redes Neurais Artificiais

2.1 Introdução

Os primeiros estudos e modelos bastante rudimentares sobre redes neurais artificiais (RNA) foram introduzidos pela primeira vez em 1943 pela celebre publicação de Warren McCulloch, no entanto, de lá para cá, várias outras foram desenvolvidas ao longo tempo, como veremos adiante.

Rede neural artificial é um preditor com recursos que permitem processar grandes volumes de informações aplicando operações matemáticas, eles são conectados entre si, um passando informação para o outro. A saída é disparada caso o valor da soma transponha a fronteira pré determinada, todo esse detalhe é inspirado no neurônio do ser humano.

Em uma das suas publicações (Haykin, 2001), define que as (RNAs) não podem solucionar problemas trabalhando de forma individual principalmente quando envolvem tarefas que requerem uma grande precisão, como aplicações de lógica e aritmética. Mas, por outro lado, as redes neurais artificiais (RNAs) têm tido um enorme sucesso em aplicações em que se exigem inferências complexas não lineares entre as entradas e saídas.

Além das qualidades já mencionadas sobre redes neurais artificiais (RNA), elas são amplamente empregadas em diversos problemas nas áreas de engenharias e ciências (Silva, Spotti, Flouzino 2016) tais como:

No ramo da saúde, com o uso das redes neurais artificiais, foi feito um rastreio de pessoas com dislexia e comorbidade de transtorno do déficit de atenção com hiperatividade (TDAH) através de classificação probabilística, com a finalidade de identificar pessoas com tais enfermidades e encaminhá-las para atendimento especializado precocemente (Mousinho et al, 2009);

No mundo militar são amplamente utilizados para localização de bases inimigas através de jatos e drones;

No automobilismo as RNA são referencias e prestam enorme contribuição no funcionamento no trafego dos carros autonomos;

São extensamente uasdos no mundo do mercado fincnceiro notadamente para a predição;

A visão computacional é uma das varias subáreas da inteligência artificial que se dedica na detecção de imagens, onde as RNAs são as referências para o desenvolvimento de modelos de algoritmos de machine learning para o reconhecimento de padrão, especificamente no reconhecimento de face, que é uma tecnica bastante consolidada. São empregue principalmente em aeroporto e rovias.

2.2 Neurônio Biológico

O neurônio humano mostardo na figura 2.1, é uma célula de aparência incomum encontrada principalmente no córtex cerebral animal (por exemplo, no meu cérebro), composto de um corpo celular englobando o núcleo e a maioria dos componentes complexos da célula e muitas extensões de subdivisões chamadas dendritos, além de uma extensão bastante longa que se chama axônio. O comprimento do axônio pode ser apenas algumas vezes mais longo do que o corpo da célula, ou até dezenas de milhares de vezes maior. Perto de sua extremidade, o axônio divide-se em muitos ramos chamados de telodendros, e na ponta desses ramos estão estruturas minúsculas chamadas terminais sinápticos ou somente sinapses que estão conectadas aos dendritos ou diretamente ao corpo celular de outros neurônios. Os neurônios biológicos recebem curtos impulsos elétricos de outros neurônios através dessas sinapses chamadas sinais. Quando um neurônio recebe um número suficiente de sinais de outros neurônios em alguns milissegundos, ele dispara seus próprios sinais (Géron A, 2019).

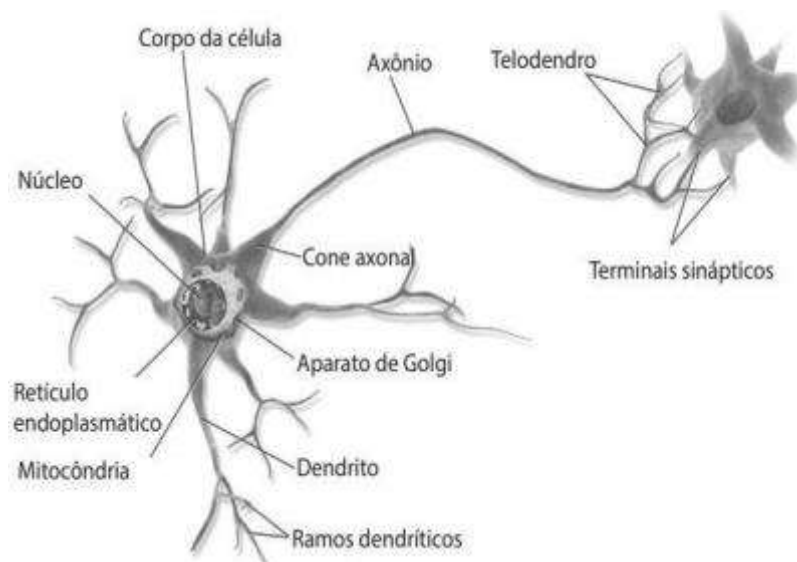


Figura 2.1 – Neurônio Biológico(Géron A, 2019)

Sendo assim, os neurônios biológicos individuais parecem se comportar de uma maneira simples, porém existem mais de 10 bilhões de neurônios no cérebro de um ser humano, que são conectados através das sinapses. Existe um enorme mistério sobre a arquitetura das redes neurais biológicas, até hoje continua

sendo objeto de pesquisa ativa, entretanto certas partes do cérebro foram mapeadas e parece que os neurônios muitas vezes são organizados em camadas consecutivas, vide a figura 2.2.

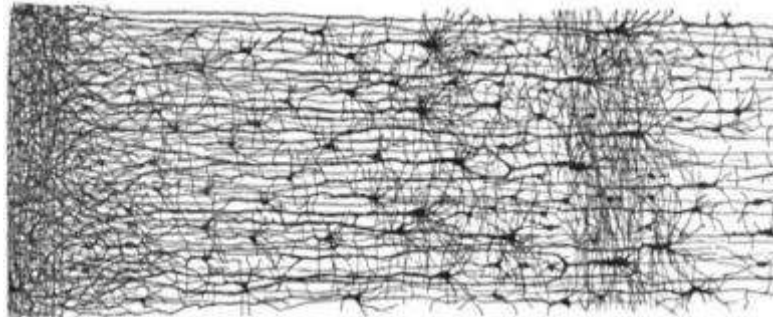


Figura 2.2 – Córtex Humano(Géron A, 2019)

2.3 Neuônio Artificial

Em 1943, Warren McCulloch e Walter Pitts definiram um modelo de neurônio artificial com base na composição e funcionalidade do neurônio humano. Esse modelo foi bem recebido pelos estudiosos no assunto, pois o funcionamento de uma rede neural artificial é bastante direto e consiste na soma dos produtos das entradas em relação aos seus respectivos pesos, seguido por uma função de ativação que define a saída, como representado na figura 2.3.

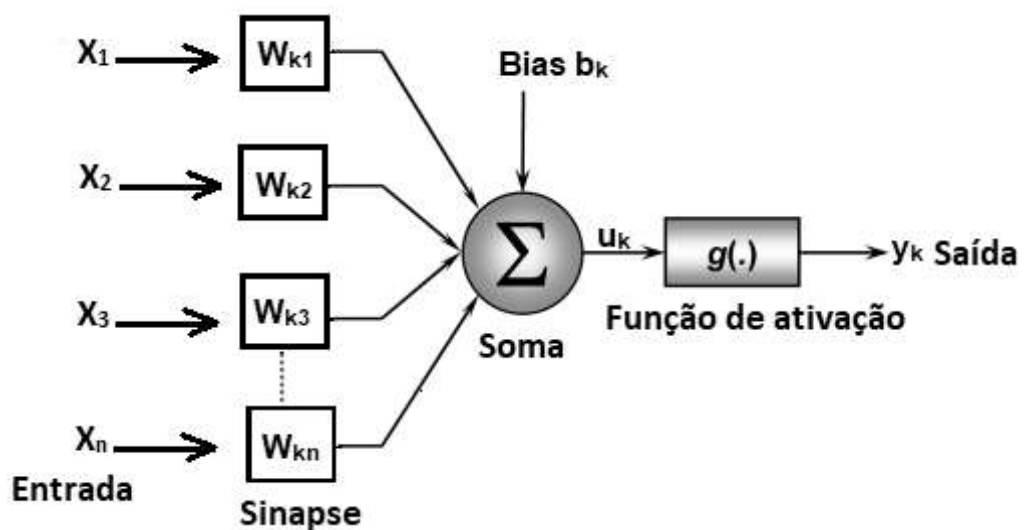


Figura 2.3 – Neuronio Artificial

Na figura 2.3, que representa um neurônio artificial, os sinais de entrada são representados pelas letras X e alimentam a rede neural artificial com dados. A excitação do neurônio depende do valor de cada peso (W). Em outras palavras, se o valor do peso for alto, a excitação será maior, caso contrário, a excitação será menor.

Dando sequência à exploração da figura 2.3, na função soma ocorre o somatório dos produtos entre os sinais de entrada e seus pesos. O resultado do somatório é a entrada da função de ativação $g(u_k)$. Se o valor de entrada ultrapassar um certo limite estabelecido, a função de ativação é ativada e produz a saída (y_k), caso contrário, a saída não é ativada.

A função de ativação tem como função limitar os sinais de saída de um neurônio dentro de um intervalo de valores pré-estabelecidos, que normalmente variam entre [0, 1] ou [-1, 1].

A expressão matemática de um neurônio pode ser descrita pelas seguintes equações:

$$y_k = g(u_k + b_k) \quad (2.1)$$

Onde:

$$u_k = \sum_{j=1}^n W_{kj} * X_j \quad (2.2)$$

Onde:

X_n = Matriz de entradas

W_k = Pesos

V_k = Resultado da soma

B_k = Bias

$g()$ = Função de ativação

y_k = Saída

2.4 Função de Ativação

A função de ativação é responsável por limitar os sinais de saída de um neurônio. A razão pela qual se utiliza a função de ativação é a capacidade que ela possui de lidar com diversos tipos de funções, sejam elas lineares ou não-lineares, introduzindo a não-linearidade no modelo. A função de ativação é responsável por tomar a decisão se um determinado neurônio artificial deve ou não ser ativado, com a finalidade de definir o sinal de saída, associado aos valores de entrada.

Existem varios tipos de função de ativação, alguns serão representado e definados no paragrafo a seguir:

A função de ativação linear possui o mesmo principio de funcionamento ao da função limiar, é uma função de ativação que define a saída do neurônio como uma combinação linear dos sinais de entrada, sem introduzir a não-linearidade no modelo. Como mostra a figura 4.

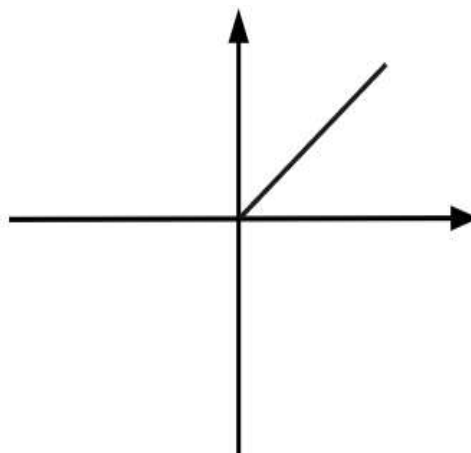


Figura 2.4 – Função Linear

A equação que define a função linear da figura 2.4 está representada a baixo:

$$g(u_k) = u_k \quad (2.3)$$

Função limiar;

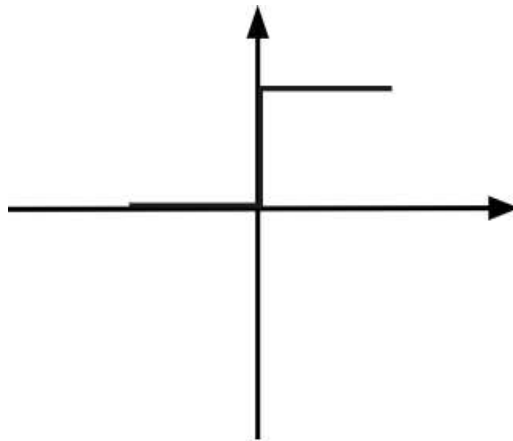


Figura 2.5 – Função Limiar

Função limiar equação 2.4:

$$g(u_k) = \begin{cases} 0, & \text{se } u_k < 0 \\ 1, & \text{se } u_k \geq 0 \end{cases} \quad (2.4)$$

A função de ativação sigmoide é uma função matemática que tem como característica principal a capacidade de transformar os sinais de entrada em uma escala de (0, 1) ou (-1, 1), tornando a saída do neurônio apropriada para análise e interpretação, são representados em diversos modelos, como veremos a seguir:

Função sigmoide logistica;

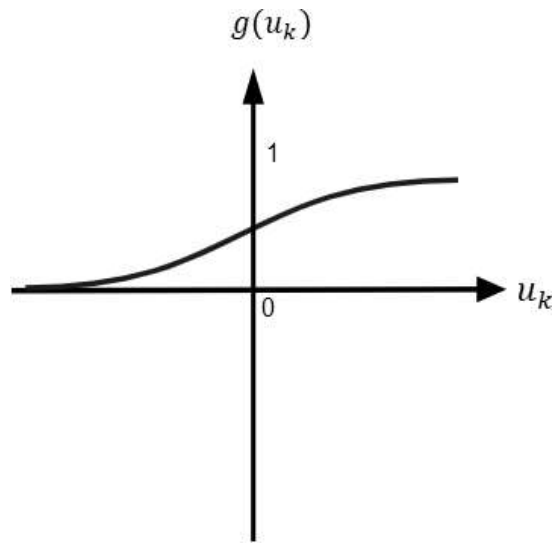


Figura 2.6 – Função sigmóide

A equação 5 representa a função sigmoide:

$$g(u_k) = \frac{1}{1 + e^{-u}} \quad (2.5)$$

Função tangente hiperbolica é uma função que transforma os sinais de entrada em uma escala de -1 a 1, fazendo com que a saída do neurônio esteja sempre dentro desse intervalo, a equação 2.6 representa essa função de ativação.

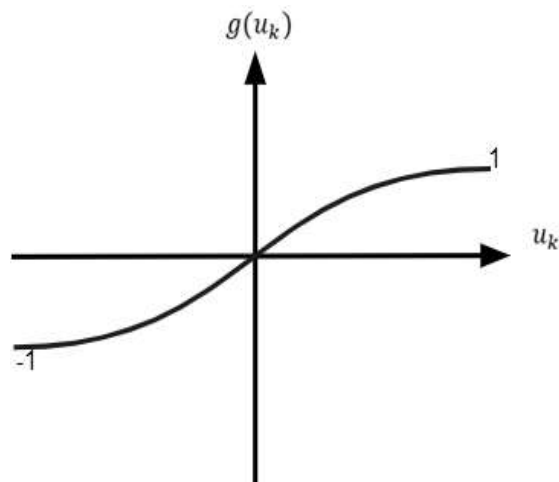


Figura 2.7 – Função tangente hiperbolica

$$g(u_k) = \frac{e^u - e^{-u}}{e^u + e^{-u}} \quad (2.6)$$

A função de ativação Relu é uma das funções mais amplamente usadas na construção de modelos de redes neurais artificiais. Ela é naturalmente fácil de interpretar, como mostrado na Figura 8. Além dos pontos já mencionados sobre essa função, ela é conveniente porque incorpora a não linearidade necessária para adaptar-se a qualquer modelo. E é uma função de ativação retificadora que define a saída do neurônio como zero para valores negativos ou o zero de entrada e mantém a saída igual à entrada para valores positivos. A função Relu é considerada uma versão melhorada da função de ativação sigmoide (logística, tangente hiperbólica).

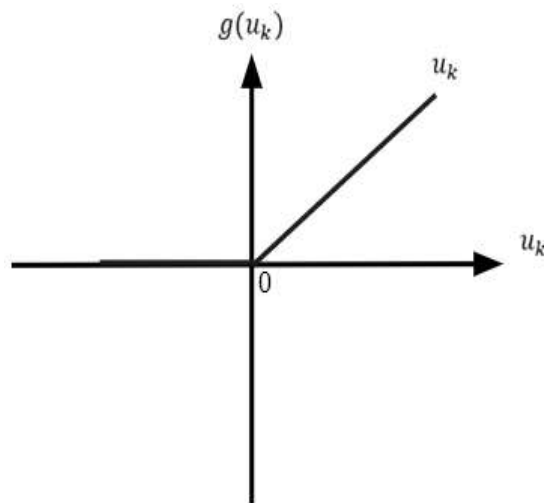


Figura 2.8 – Função de ativação Relu

Equação da função de ativação *ReLU*

$$g(u_k) = \begin{cases} 0, & \text{se } u_k \leq 0 \\ u_k, & \text{se } u_k > 0 \end{cases} \quad (2.7)$$

2.5 Arquitetura

A tarefa de descrever como um ou vários problemas devem ser definidos e resolvidos carece de um rigoroso planejamento, e em várias situações, normalmente recorremos a softwares computacionais para facilitar o processo de resolução.

Desta feita, é fundamental a definição da arquitetura do modelo da rede, pois é através dela que definimos como os neurônios deverão comportar-se uns em relação aos outros, a fim de estabelecer uma conexão bem estruturada entre as camadas e os neurônios. Na literatura, são estudados vários tipos de arquitetura de RNA, que serão apresentados no parágrafo a seguir com mais detalhes:

2.5.1 Arquitetura Direta (*feedforward*)

Essa arquitetura é mostrada na figura 9, é um modelo que representa uma rede neural simples, de sentido único, e é composto por apenas uma camada de neuronios, as variáveis de entrada são recebidos pela camada de entrada, que de forma sequencial fazem a conexão dos sinais com a única camada de neurônios ate a saída.

Uma das suas características principais nesse tipo de rede, a informação flui em uma única direção, da entrada para a saída ou seja da esquerda para direita.

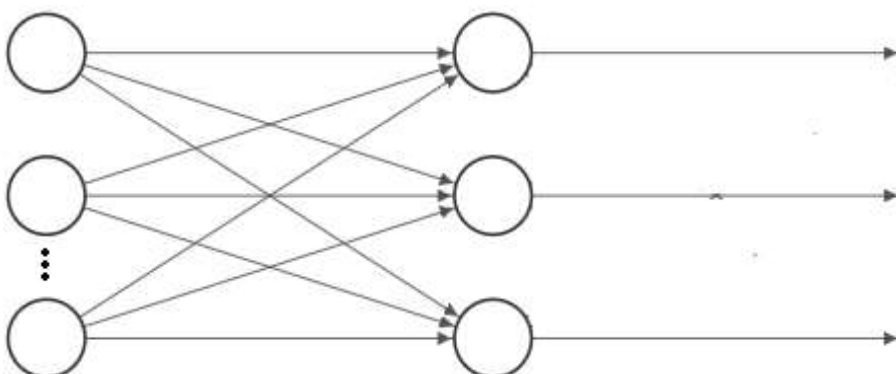


Figura 2.9 – Camada Direta

Para situações linearmente separáveis esse modelo funciona perfeitamente. Elas são comumente usadas em situações em que haja necessidade de classificações e regressão de padrões e problemas lineares.

2.5.2 Arquitetura Direta de Múltiplas Camadas

Nesta arquitetura, o sentido do fluxo também é unidirecional. No entanto, ela difere da arquitetura 2.5.1 por conter uma ou várias camadas escondidas entre as camadas de entrada e saída. Por serem mais robustas em relação à arquitetura anterior, geralmente são empregadas para resolver múltiplos problemas que apresentam não-linearidades, melhorar processos de produção, robótica, procedimentos de manufatura e hospitalares (Da Silva et al, 2010).

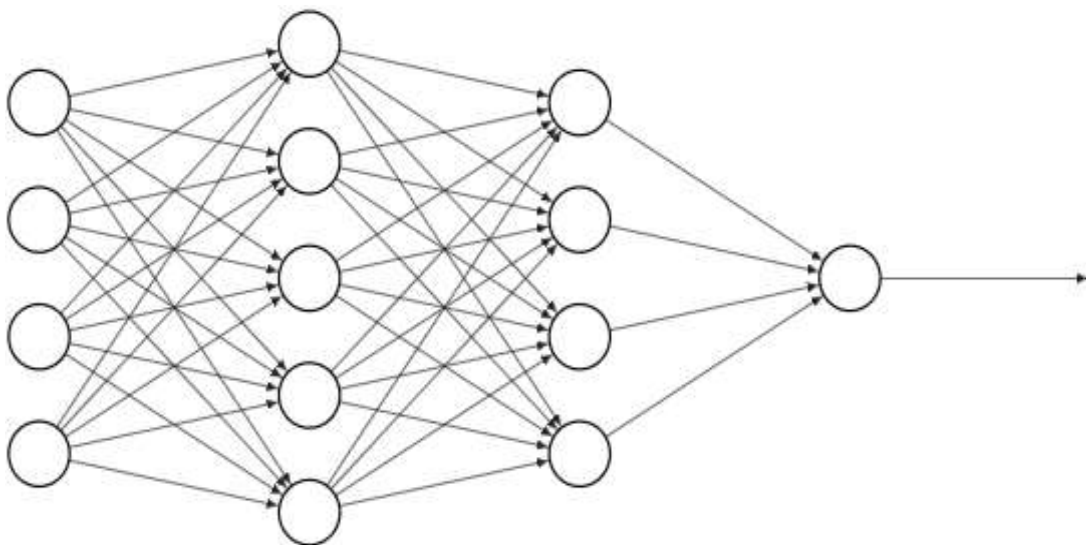


Figura 2.10 – Múltiplas Camadas

Na figura 10 está representada uma arquitetura de Múltiplas Camadas (*feedforward*), que conta com uma camada de entrada. O seu princípio de funcionamento é similar ao da figura 9 em termos do sentido do deslocamento do fluxo de informação. No intervalo entre a camada de entrada e a de saída, existem neurônios que compõem duas camadas escondidas que se conectam

entre si até a camada de saída. As camadas escondidas estão conectadas uma após a outra e recebem sinais da camada de entrada. Esses mesmos sinais de entrada não sofrem nenhuma alteração ao longo do processo. A segunda camada escondida recebe sinais de saída da primeira camada escondida e produz o resultado completo em direção ao modelo de ativação abastecida pelas ligações da fonte localizadas na entrada. Cada neurônio possui seu respectivo peso em cada uma das suas entradas e uma polarização (Grus J, 2016).

No entanto, devido à sua multiplicidade em termos de camadas, essas redes neurais podem ser aplicadas em uma vasta área de conhecimento, incluindo reconhecimento de padrões, agrupamento de dados brutos, estimativa de temperatura, previsões no mercado financeiro e muito mais.

2.5.3 Arquitetura de Rede Recorrente (Realimentação)

São redes que, em suas arquiteturas, possuem laços para o processo de realimentação, ou seja, laços de repetição. Elas são diferentes das duas mencionadas anteriormente tanto em sua configuração quanto em seu funcionamento. Além disso, apresentam um dinamismo próprio, pois não se limitam a considerar apenas os exemplos de entrada vistos naquele instante, mas também utilizam os sinais de saída para retroalimentar a entrada principal de forma a interferir positivamente no resultado, como ilustrado na figura 2.11.

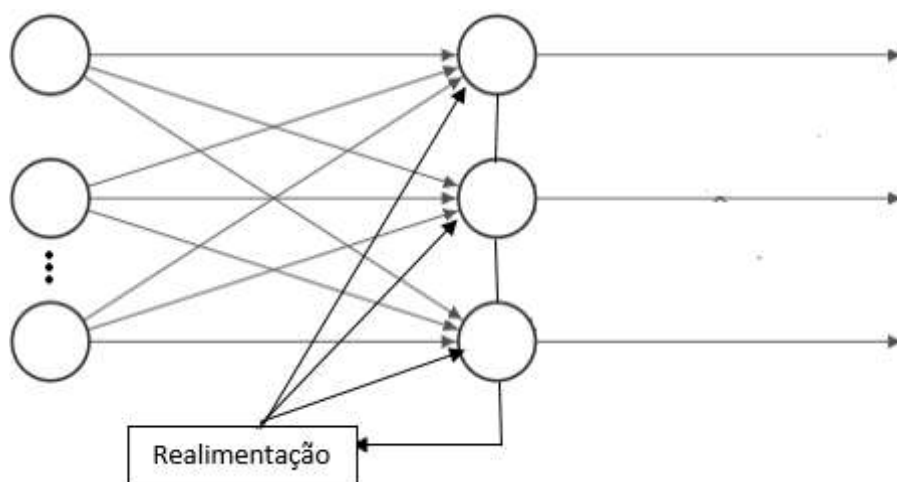


Figura 2.11 – Rede Recorrente

A escolha de uma rede recorrente no momento $t-1$ afeta a decisão que será tomada em um momento posterior, na etapa de tempo t . Portanto, é correto afirmar que esses modelos possuem mais de uma fonte de alimentação em seu funcionamento: a entrada inicial e a sua saída, que se combinam para definir como a rede irá responder a novos dados (Da Silva et al, 2010).

Exemplo de redes que possuem a realimentação são: a rede Hopfield, rede Perceptron com realimentação, rede Kohonen.

2.6 Aprendizado de Máquina

Uma das qualidades das redes neurais é a capacidade que elas possuem de aprender por meio de padrões, permitindo que sejam adaptadas para resolver diversos problemas. O processo de aprendizado ocorre de forma interativa e ordenada, com a atualização das sinapses e seus níveis, como o bias, a fim de tornar a rede apta para ser usada de forma geral. Essas interações realizadas de maneira ordenada deixam a rede mais instruída para qualquer ambiente de processo de aprendizagem, a que se dá o nome de algoritmo de aprendizagem (Haykin, 2001).

Existem vários tipos de aprendizado de máquina, portanto, é fundamental mencioná-los, pois estão categorizados levando em conta suas especificidades, como a predição, as características dos dados, o modelo e o tipo de supervisão, de acordo com a quantidade (Géron A, 2019).

Existem diversos modelos e categorias de tipo de aprendizado, a seguir citaremos alguns:

2.6.1 Aprendizagem Supervisionada

O aprendizado supervisionado, também chamado de aprendizado com professor, ocorre quando o modelo aprende a partir de resultados esperados já

pré-estabelecidos, ou seja, cada exemplo de entrada é associado a uma saída desejada, utilizando os valores passados da variável dependente para aprender quais devem ser seus resultados de saída. Os mesmos valores servem como supervisão dessas previsões, permitindo o ajuste nas previsões com base nos erros. Um exemplo muito famoso de aprendizado supervisionado é o filtro de spam, como mostrado na figura 2.12. Quando recebemos mensagens via email, o destino dado a essa mensagem é usado como parte do treinamento pelos algoritmos. Dessa forma, os emails recebidos e lidos com frequência são entendidos como mensagens importantes e sempre serão direcionados para a caixa de mensagens recebidas (INBOX). Por outro lado, quando certas mensagens direcionadas para a caixa de entrada são constantemente ignoradas, serão então deduzidas como mensagens irrelevantes ou maliciosas, tendo como destino as caixas de spam ou lixeira (Géron A, 2019).

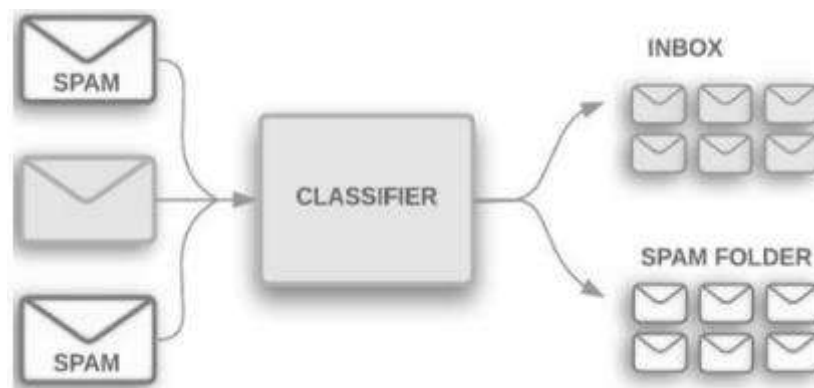


Figura 2.12 – Aprendizado Supervisionado Adaptado [2]

O procedimento de identificação e separação quanto à importância das mensagens recebidas é realizado por um classificador de padrões. As próximas mensagens passarão por um filtro com o propósito de classificá-las tendo como referência as condições de spam ou não spam. Se a mensagem for considerada spam, será direcionada para a caixa de spam (SPAM), caso contrário, irá para a caixa de entrada (INBOX).

2.6.2 Aprendizado não Supervisionado

Este tipo de aprendizado é conhecido como aprendizado não supervisionado. Nele, as variáveis de entrada são amostradas sem informações sobre a variável alvo ou rótulo, ou seja, não há valores rotulados pré-estabelecidos que possam ser usados como modelo de referência para fazer previsões futuras de forma comparativa. No entanto, nesse sistema de aprendizado, o modelo se ajusta com certa frequência aos sinais que recebe, aprendendo aptidões que lhe possibilitam desenvolver competências.



Figura 13 – Blocos de aprendizado não supervisionado

A aprendizagem competitiva é uma técnica de aprendizado não supervisionado em que vários neurônios competem entre si para serem ativados pela entrada de dados, de forma que apenas um neurônio vence e é ativado enquanto os demais são inibidos. Essa técnica é frequentemente utilizada em redes neurais de agrupamento, onde o objetivo é agrupar dados similares em um mesmo cluster. Quando os neurônios competem para serem ativados, acabam agrupando os dados em diferentes clusters, dependendo de suas características. A partir desses clusters, é possível identificar padrões nos dados e gerar informações úteis.

2.6.3 Aprendizado por Reforço

A Aprendizagem por Reforço é baseada em sistemas de recompensa, onde a máquina por si só deve solucionar problemas por tentativa e erro até que o sistema aprenda qual é a melhor sequência de tomada de decisão. Nesse tipo de aprendizado, a máquina aprende com seus próprios erros, buscando sempre maximizar a recompensa obtida a cada ação realizada.

Neste método, os parâmetros internos dos neurônios são baseados em qualquer informação, seja ela com qualidade ou sem qualidade, em grandes volumes ou em menor volume, provenientes do processo de interação com o ambiente. Tem como finalidade testar a dinâmica do aprendizado (da Silva et al, 2010)

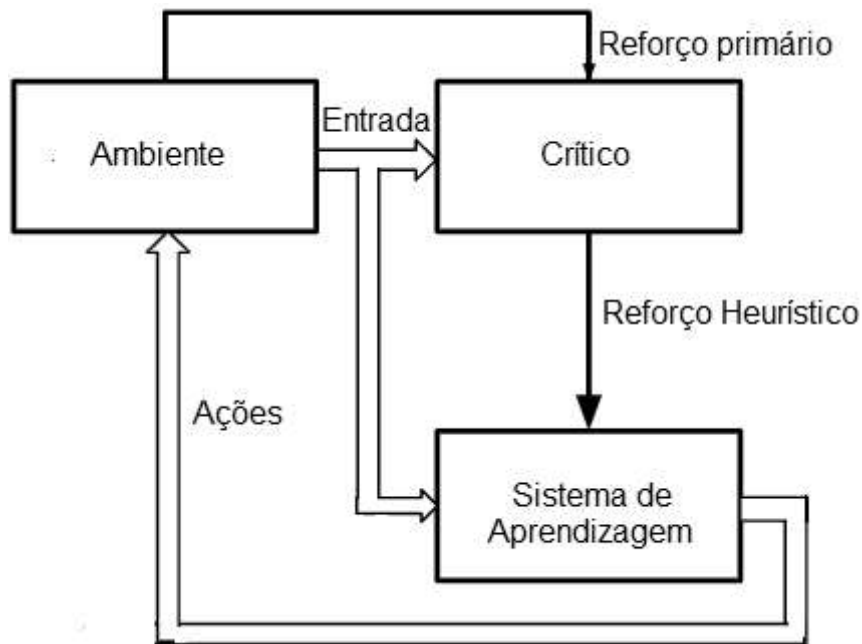


Figura 2.14 – Aprendizagem por Reforço

Na figura 2.14, é representado um modelo de aprendizagem por reforço em diagrama de blocos. O sinal de reforço primário proveniente do ambiente é transformado pelo crítico em um sinal de reforço de melhor qualidade, chamado de reforço heurístico. (Haykin, 2009)

Nesse modelo, tem-se como propósito atenuar significativamente o total de erros, tendo como premissa a aprendizagem por reforço, que é largamente baseada em metodologias estatísticas, em que os ajustes são realizados de forma probabilística. Caso os resultados sejam corretos, o modelo é recompensado.

2.7 Aprendizagem

Agora que já foram mostrados com detalhes sobre os tipos de aprendizado, chegou o momento de definir e atribuir alguns parâmetros essenciais, pois, salientar ainda que a definição do modelo do algoritmo a ser aplicado, está estreitamente ligado ao trabalho de aprendizagem que se deseja desenvolver com a RNA. No próximo parágrafo serão definidos alguns parâmetros de aprendizagem (Haykin, 2001):

2.7.1 Associação de Padrões

Existem essencialmente dois tipos de associações, a auto-associação e a heteroassociação. Na auto-associação uma coleção de vetores tem de ser armazenado em um modelo de RNA, e posteriormente, apresentados de forma sequencial ao referido modelo, no entanto, com um formato truncado. É um momento em que o modelo necessita resgatar os padrões relatados anteriormente, por intermédio de coleções até gerar produzir uma solução adequada. No segundo caso, a heteroassociação é o inverso da auto-associação o funcionamento é semelhante ao tipo de aprendizado com professor, onde os sinais de entrada têm uma relação com os de saída.

2.7.2 Reconhecimento de Padrões

É um padrão relativamente complexo e muito eficaz, possui a vantagem de poder ser usado para os dois tipos de aprendizado, o supervisionado e o não supervisionado através de processo de treinamentos. A rede neural aprende padrões pre estabelecidos, recebendo vários sinais de entrada de forma reiterada, atrelados as entradas estão os sinais de referências pertencente a cada sinal de entrada.

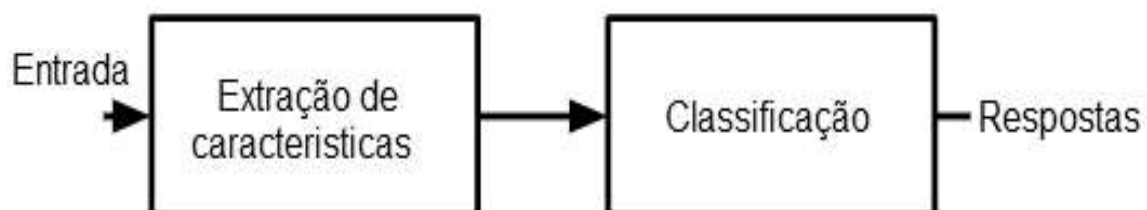


Figura 2.15 – Classificação de padrões

Em seguida, acontece o processo de testagem onde o modelo recebe dados dos quais a rede desconhece, no entanto, com as mesmas características dos dados usados no ato de treinamento. Desse modo a rede está apta para reconhecer a classe de cada padrão específico, isso só é possível através das informações extraídas no processo de treinamento.

2.7.3 Aproximação de Funções

O processo de aproximação consiste no treinamento do modelo de rede neural, tendo como premissa os dados de entrada e os de saída. A finalidade é projetar um modelo de RNA de forma a interligar o sistema desconhecido para apresentar detalhadamente o processo de entrada e saída para averiguar se sistema de aproximação está funcionando corretamente.

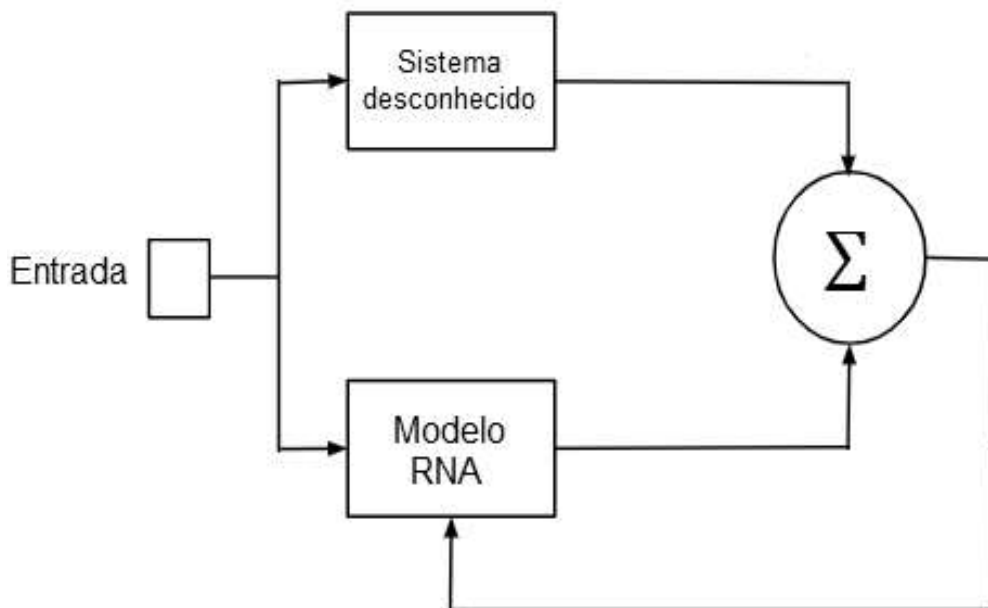


Figura 2.16 – Função de aproximação

Onde o vetor entrada é representado pela letra x , já o $f(x)$ designa o sistema desconhecido e a d descreve a saída desejada, o que representa o sistema de aprendizado com professor como mostra a figura 16.

2.7.4 Coontrole

Outra área onde as RNAs são bastante eficientes é no sistema de controle como mostra a figura 2.17, tem um funcionamento identico ao aprendizado supervisionado, pois, existe um sinal pré estabelecida sendo usado como referência onde a saída realimenta a entrada principal.

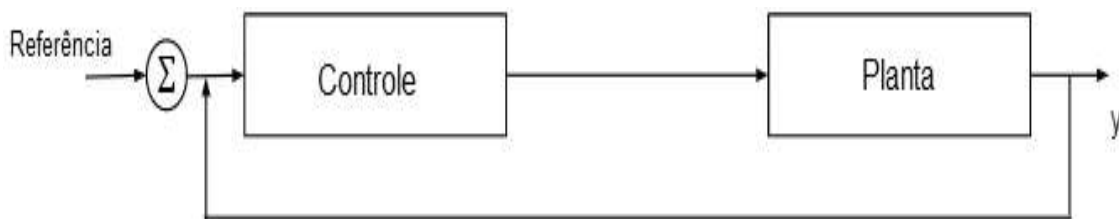


Figura 2.17 – Sistema de controle

Com o sinal pré estabelecido ou de referência analisa-se a direferença entre a referencia e o valor de saída, então, o resultado gerado pela diferença é submetido ao controle do modelo para que seja moldado dentro dos parâmetros adequados, com o intuito de garantir que os valores de entrada sejam os mais precisos possiveis para que seja evitado erros nos sinais de saída.

2.8 Perceptron de Única Camada

O Perceptron é a forma mais simples de representação de uma RNA, constituída somente por uma unidade neural desenvolvida por Rosenblatt (1958), cuja expectativa era implementar um modelo computacional inspirado na retina (Da Silva et al, 2019). Por este motivo, em algumas literaturas, também é chamado de perceptron de Rosenblatt. A rede perceptron é constituída por apenas uma camada, daí vem a sua simplicidade, e nessa única camada contém apenas um neurônio artificial (NA).

A figura 2.18 ilustra o perceptron, que recebe sinais de entrada da área externa ($x_1...x_n$), para ilustrar a situação na qual se deseja controlar. O modelo possui

uma única saída binária, pois é composto por somente um único neurônio. Essa simplicidade permite acelerar todo o processo de treinamento e torna-o capaz de solucionar somente problemas lineares.

Apesar de ser um modelo simples, na época de seu desenvolvimento, o Perceptron teve um enorme potencial de atrair diversos pesquisadores interessados em investigar essa área de pesquisa promissora para a época. Ele também chamou a atenção da comunidade científica que trabalhava com inteligência artificial. (Da Silva et al, 2019).

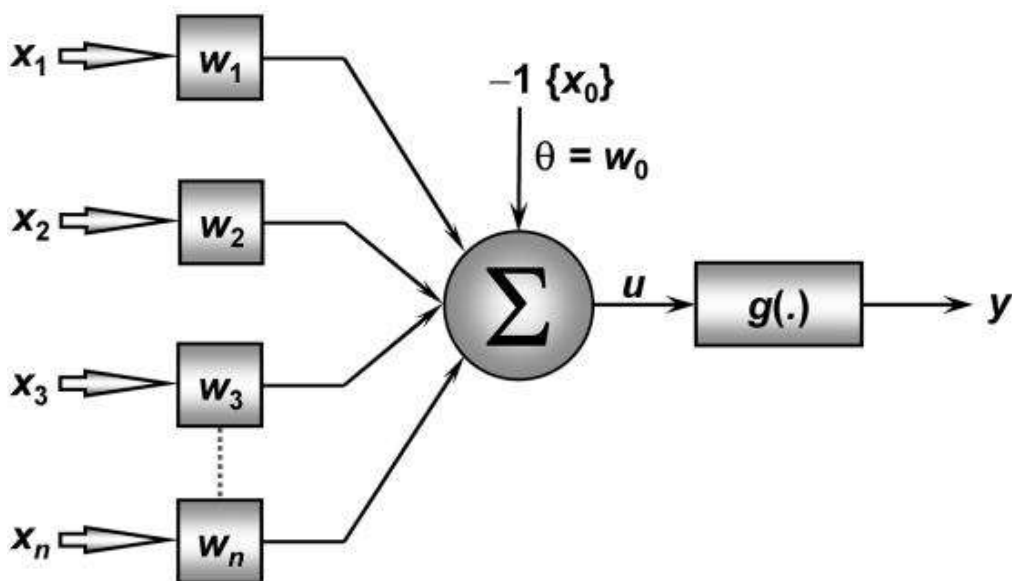


Figura 2.18 – Rede perceptron (Da Silva)

O perceptron da figura 2.18 faz parte de um modelo *feedforward* atendendo o sentido do tráfego das informações têm um único sentido, que são da esquerda para direita em direção a saída do perceptron sem a possibilidade de retroalimentação.

2.8.1 Funcionamento do Perceptron

A estrutura do modelo, como representado na figura 2.18, é de fácil interpretação. As entradas X_i representam os sinais que descrevem o desempenho do desenvolvimento da estrutura que será mapeada, e essas entradas são multiplicadas pelos seus respectivos pesos W_i , que indicam o quão importante cada sinal é para o modelo. É importante ressaltar que os sinais de entrada são imutáveis e podem adotar diversos tipos de valores, enquanto os pesos sinápticos são inicialmente determinados de forma aleatória ou randômica.

Em seguida, a soma dos produtos entre as entradas e seus pesos ponderados, adicionado ao respectivo bias θ , é então enviado para a função de ativação, com a finalidade de determinar o sinal de saída y do modelo.

As equações 2.8 e 2.9, representam matematicamente o funcionamento de perceptron:

$$\begin{cases} u = \sum_{i=1}^n W_i * X_i - \theta \\ y = g(u) \end{cases} \quad (2.8)$$

$$(2.9)$$

Em que:

X_i representam os sinais de entradas do modelo, o W_i são os pesos relacionados a entrada

X_i = Matriz de entradas

W_i = Pesos

u = Potencial de ativação

θ = Bias

$g()$ = Função de ativação

y = Saída

Para perceptron simples como o da figura 2.18, é comumente utilizado a função de ativação que pode solucionar problemas linearmente separáveis, pois, nesses casos os resultados a serem produzidos para a saída do modelo são binários, 0 ou 1.

2.8.2 Treinamento do Perceptron

Baseando-se na regra de Hebb, o processo de treinamento de um perceptron simples consiste basicamente na atualização dos pesos sinápticos. Devido à sua simplicidade, os valores alvo que representam os sinais a serem atingidos fazem parte de um conjunto de classes binárias, normalmente 0 ou 1.

De forma breve, o processo de treinamento consiste basicamente no seguinte: quando o resultado da execução do modelo não for idêntico ao valor alvo, é iniciada a atualização. Como mencionado anteriormente, os valores de entrada são imutáveis, ou seja, não podem ser alterados. Portanto, o processo de atualização é feito nos pesos, alterando de maneira crescente os seus valores. Mas se o resultado da saída do modelo coincidir com o valor alvo, indica que os pesos devem permanecer intactos sem necessidade de atualizá-los. Então, este procedimento é reproduzido repetidamente com todos os sinais de entrada do conjunto de dados até que a saída do modelo seja igual ao alvo a fim de concretizar o objetivo. Matematicamente, o processo de atualização dos pesos é representado da seguinte forma:

$$W_i^{atual} = W_i^{anterior} + \eta \cdot (d^{(k)} - y) \cdot X_i^{(k)} \quad (2.10)$$

Onde:

W = Pesos sinápticos

η = Taxa de aprendizagem (constante)

$d^{(k)}$ = Sinal alvo ou saída desejada

y = Saída do modelo

$X_i^{(k)}$ = Sinal de entrada

2.9 Perceptron Multicamada

Os perceptrons multicamadas foram introduzidos inicialmente na década de 1980 com grande contribuição dos autores McClelland e Rumelhart (1986), que definiram de forma exaustiva o modelo de aprendizagem backpropagation, tornando-se um grande sucesso para as RNAs.

O Perceptron Multicamadas (também conhecido como MLP - Multi-Layer Perceptron) é uma arquitetura melhorada se comparada ao perceptron simples. Ele é um modelo de RNA que contém uma ou mais camadas intermediárias entre a camada de entrada e a camada de saída. Essas camadas intermediárias são compostas por neurônios que processam os sinais de entrada e produzem saídas que são passadas para a próxima camada até que a saída final seja produzida.

O processo de treinamento do Perceptron de multicamadas é uma generalização da regra Delta, que se assemelha parcialmente ao processo de treinamento do Perceptron de uma camada e ao do modelo Adaline. Resumidamente, o sistema de retropropagação é a base principal do processo de aprendizado do Perceptron de multicamadas.

Devido às qualidades já mencionadas no primeiro parágrafo, os perceptrons multicamadas vêm sendo empregados em tarefas de variadíssimas naturezas e complexidades, fazendo uso de sua capacidade de não-linearidade, o que lhes permite apresentar resultados robustos e de qualidade. Algumas das áreas que se beneficiam desses modelos são:

Detecção facial;

Detecção de voz;

Detecção de doenças;

Controle desmatamento florestais;

Previsão do tempo.

A Figura 2.19 representa a etapa de forward do movimento do perceptron multicamadas, em que a sequência de dados tem início na camada de entrada, passa pelos neurônios da primeira camada escondida e da segunda camada escondida, e percorre o modelo até alcançar a saída, da esquerda para a direita.

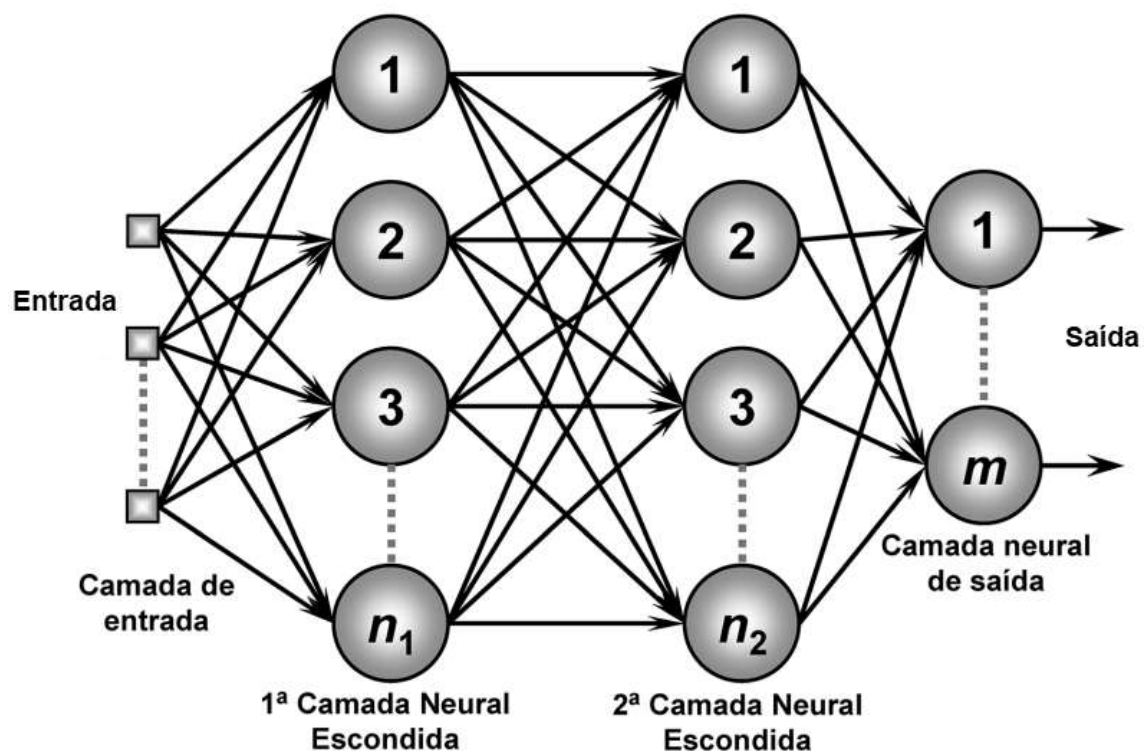


Figura 2.19 – Perceptron Multicamadas adaptado(Da Silva)

2.9.1 Funcionamento de Perceptron Multicamadas

Conforme demonstrado na Figura 2.19, as fontes externas que alimentam a rede, representando as entradas, seguem o fluxo da informação no sentido convencional até alcançar a saída do modelo. A primeira camada intermediária da RNA recebe sinais vindos da entrada, filtrados pelas respectivas funções de

ativação, que determinam os valores da saída da primeira camada intermediária. Ainda é importante salientar que as saídas da primeira camada intermediária passam a alimentar os neurônios da próxima camada. Em algum momento do processo, as saídas de um determinado neurônio serão consideradas como entrada do próximo, até que seja alcançada a saída final do modelo.

Posto isto, o número de neurônios contidos na camada escondida não alterará a trajetória do fluxo das informações, que, para o *forward*, é e sempre será da esquerda para a direita.

2.9.1 Treinamento do Perceptron multicamadas

O processo de treinamento da RNA de multicamadas é baseado na generalização

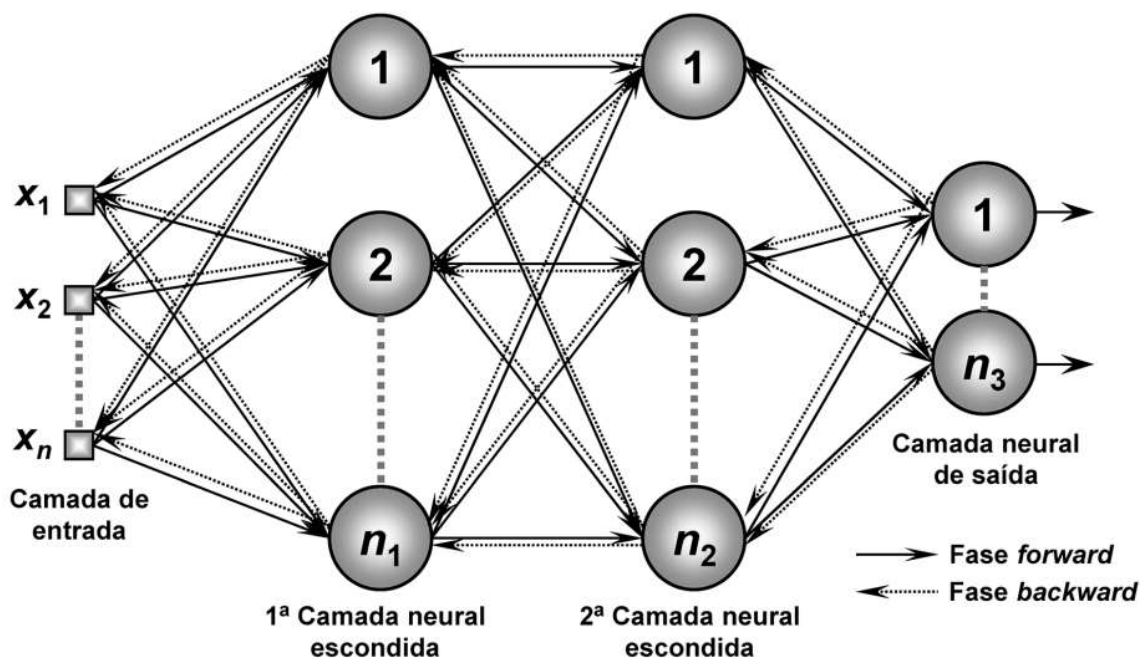


Figura 2.20 – Perceptron Multicamadas Adaptado(Da Silva)

Todavia, inicialmente é feito o processo de propagação (forward), como detalhado na Figura 2.19, que representa o sentido normal do fluxo de informação da rede, através do qual os sinais $\{x_1, x_2, \dots, x_n\}$ que representam as

entradas da rede se propagam entre os neurônios intermediários até alcançarem a camada neural de saída. O forward tem como objetivo apresentar os resultados finais das saídas realizadas pelo modelo de RNA.

Em seguida, os resultados previstos pelo modelo de RNA são instantaneamente comparados com os sinais alvo esperados, seguindo a regra de aprendizagem supervisionada. Caso a resposta fornecida pela rede satisfaça a referência, entende-se que o modelo aprendeu perfeitamente durante o treinamento. É importante salientar que é praticamente impossível obter um resultado satisfatório logo no primeiro treinamento ou nas primeiras épocas.

A diferença entre as respostas desejadas e os resultados produzidos pelos neurônios de saída da rede é o que chamamos de erro, resultante do processo de interação pelo qual o modelo foi submetido. Nesse momento, termina temporariamente a sequência direta (*forward*) do treinamento, dando início ao próximo passo, que é o ajuste das sinapses e limiares, tecnicamente chamado de processo inverso do primeiro passo, o *backpropagation* (propagação reversa), percorrendo todas as camadas da rede. Para facilitar o entendimento desses procedimentos, a figura 2.20 apresenta mais detalhes. O objetivo desses ajustes é melhorar a performance do modelo e reduzir significativamente a quantidade de erros, aproximando-a de zero.

No que diz respeito aos ajustes das sinapses dos neurônios, são definidas suas componentes, que determinam as conexões entre neurônios contidos nas camadas que constituem o modelo, como demonstrado na Figura 2.21, a qual define os elementos dentro do modelo e a forma como eles se relacionam com diferentes variáveis.

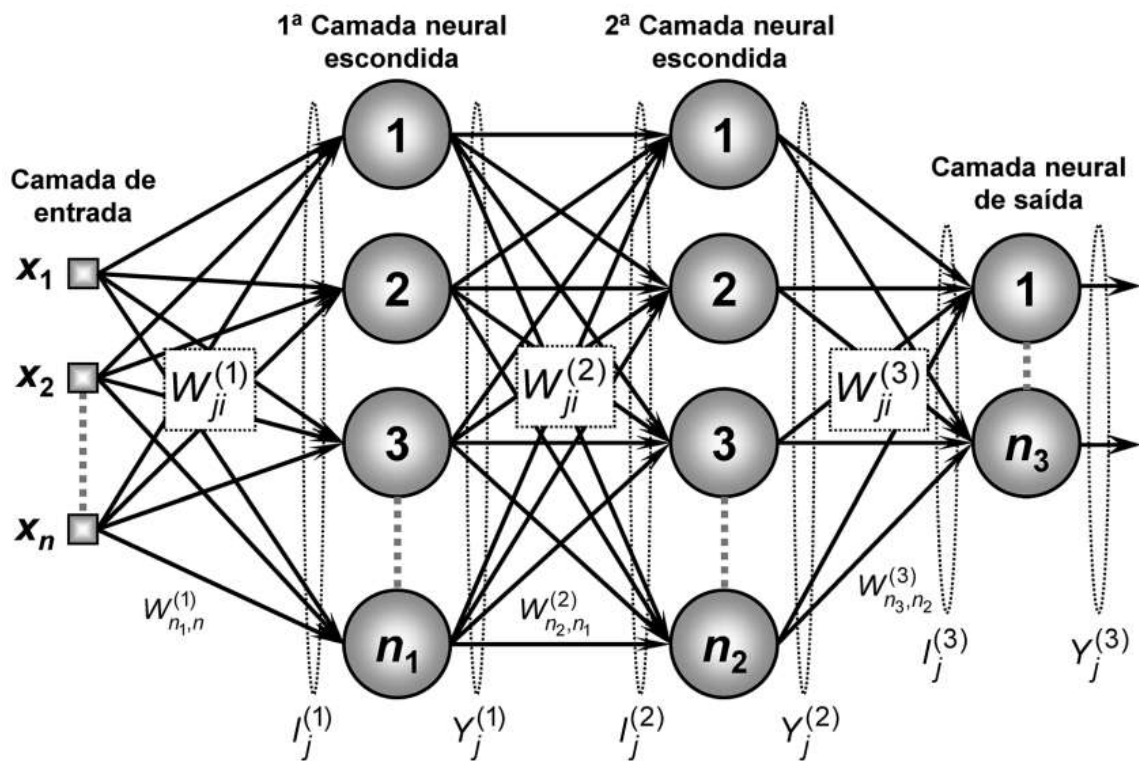


Figura 2.21 - Perceptron Multicamadas(Da Silva)

É necessário fazer uma analogia entre as Figuras 2.21 e 2.22 para facilitar o entendimento da operação do *backpropagation*, pois os neurônios das camadas intermediárias e da camada de saída são representados na Figura 2.22 pelo símbolo j , o qual pode ser qualquer um dos neurônios da Figura 2.21. Por sua vez, o símbolo L representa qualquer camada, também na Figura 2.21, enquanto que g pode ser qualquer um dos modelos de função de ativação descritos na Seção 2.4 e contidos nos neurônios intermediários e de saída.

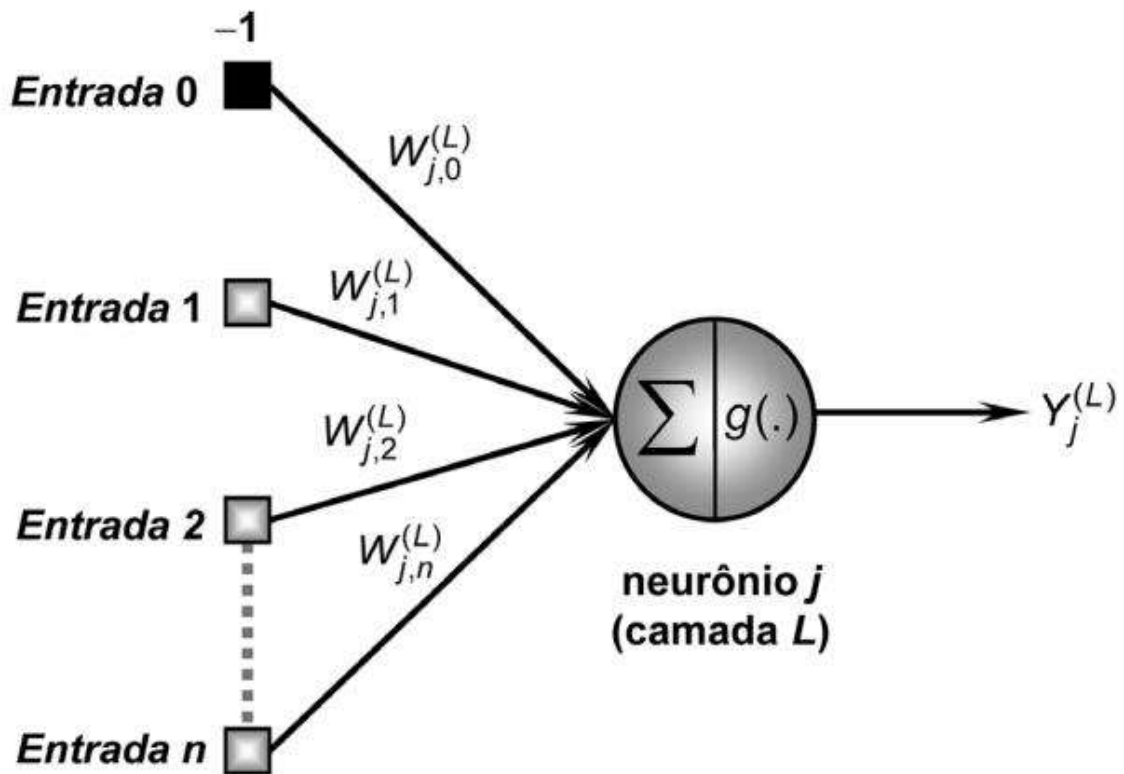


Figura 2.22 Representação da retropropagação

Onde:

W_{ji} = Pesos sinápticos

I_j = O somatório dos produtos entre as entradas e seus respectivos pesos sinápticos

y_j = Saída do modelo

x_n = Sinal de entrada

Atentando-se especificamente à Figura 2.22, que remete a um perceptron de uma camada, como visto no subcapítulo 2.8, observa-se que ela é uma representação simplificada de um perceptron de multicamadas, no qual as entradas são representadas pelos neurônios das camadas intermediárias da Figura 2.21, ponderados pelas respectivas sinapses, e em seguida submetidos à função de ativação, seguindo o fluxo sequencialmente até a saída. Essa

descrição detalha a fase *forward* do *perceptron* de multicamadas. Com base nas estruturas das arquiteturas dos modelos de RNA nas figuras 2.20, 2.21 e 2.22, o treinamento da rede *backpropagation* é dividido em duas partes distintas: a parte *forward* e a parte *backward*, que serão detalhadas no parágrafo a seguir.

- O primeiro passo consiste no processo da inicialização das sinapses W , onde, os valores são determinados randomicamente e normalmente próximos de zero.
- Em seguida, o modelo recebe as amostras nas camada de entrada, por se tratar de uma sistema supervisionado, então, também recebe as amostras referente aos sinais alvos esperados. O processo de aprendizagem é do tipo *off-line* que consiste em analisar enormes quantidades de dados e executa o treinamento de forma *off-line*, é um tipo bastante lento na execução além do alto consumo de recursos computacionais, porém garante maior precisão na apuração que reflete positivamente na acuracia da saída da RNA.

A equação 2.11 representa o início do processo de treinamento, através da soma ponderada das entradas e pesos que são alimentados à função de ativação, resultando na saída que determina o erro gerado pela rede. É importante ressaltar que a primeira camada de entrada não passa pelo processo iterativo, ou seja, não contém função de ativação entre seus neurônios

$$I_i = \sum_{i=0}^n W_{ji} \cdot X_i \quad (2.11)$$

Durante o treinamento do *perceptron* de multicamadas da figura 2.21, em algum momento, as camadas intermediárias assumirão o papel de camadas de entrada. Nesse caso, a equação 2.11 deve ser substituída pelas equações 2.13 e 2.14, que são usadas para calcular as saídas das camadas intermediárias e a camada de saída, respectivamente.

$$I_j^{(1)} = \sum_{i=0}^n W_{ji}^{(1)} \cdot X_i \quad (2.12)$$

$$I_j^{(2)} = \sum_{i=0}^{n_1} W_{ji}^{(2)} \cdot Y_i^{(1)} \quad (2.13)$$

$$I_j^{(3)} = \sum_{i=0}^{n_2} W_{ji}^{(3)} \cdot Y_i^{(2)} \quad (2.14)$$

Em que o n ilustra a quantidade de valores de entradas submetidos aos neurônios.

No entanto, o $Y_j^{(L)}$ é a representação na forma vetorial das informações da saída do modelo de j -ésimo neurônio com base no neurônio da figura 2.22. Como mostrados nas equações a seguir.

$$Y_j^{(1)} = g(I_j^1) \quad (2.15)$$

$$Y_j^{(2)} = g(I_j^2) \quad (2.16)$$

$$Y_j^{(3)} = g(I_j^3) \quad (2.17)$$

$Y_j^{(1)}$ é a saída da primeira camada intermediária

$Y_j^{(2)}$ é a saída da segunda camada intermediária

$Y_j^{(3)}$ é a camada de saída do modelo da rede neural

(I_j^L) é a função de ativação contida em todos os neurônios, e é o que define que tipo de resultado deverá ser reproduzido pela saída y_j^L da RNA.

É importante resaltar mas uma vez, que, com base na figura 2.21 a saída da primeira camada intermediária, no sentido convencional do treinamento alimenta a entrada da segunda camada intermediária, e esta, alimenta a camada de saída.

Até o momento, vimos apenas o trecho da derivação referente ao forward, que representa a primeira parte do treinamento. A segunda etapa é o *backpropagation*, que consiste em detalhar o conceito de erro, cuja responsabilidade é analisar a diferença entre o resultado encontrado pela saída do modelo da RNA e os sinais de referência pré-estabelecidos ou valor alvo pretendido. Logo, com base na figura 2.21 e levando em conta os sinais de referência *k-ésimos*, a equação 2.18 ajuda a definir a métrica utilizada para estimar a qualidade do desempenho do modelo da RNA através dos sinais da camada de saída, aplicando a função *mean squared error*.

$$E(k) = \frac{1}{2} \sum_{j=1}^{n_3} (d_j(k) - Y_j^3(k))^2 \quad (2.18)$$

Em que:

$Y_j^3(k)$ é o sinal da camada de saída do modelo de RNA da figura 2.21, mediante *k-ésimo* de processo;

$d_j(k)$ corresponde ao valor alvo

A equação 2.18 define a media dos erros também pode ser representada da seguinte maneira:

$$E_M = \frac{1}{2} \sum_{k=1}^p E(k) \quad (2.19)$$

E_M Mean Squared Error

$E(k)$ Squared Error

O desenvolvimento do *backpropagation* compreende a atividade de atualização dos pesos W com a finalidade de reduzir significativamente o resultado da equação 2.18, com isso, melhorar a eficácia da saída da RNA.

- A segunda parte do processo de treinamento compreende na a atualização das sinapses da RNA referente a fase do *backward*.

O ajuste das sinapses $W^{(3)}$ dos neuronios da camada de saída da figura 21, é baseado na regra do gradiente descendente que define o tempo do percurso para alcançar o minimo local, e está consiste na análise e implementação das propriedades derivativas parciais as quais garantem a descida e o sentido do gradiente, baseado na função erro através da equação 2.18.

$$\nabla E^{(3)} = \frac{\partial E}{\partial W_{ji}^{(3)}} = \frac{\partial E}{\partial Y_j^{(3)}} * \frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}} * \frac{\partial I_j^{(3)}}{\partial W_{ji}^{(3)}} \quad (2.20)$$

Por outro lado, e com base na equação anterior, associa-se:

$$\frac{\partial I_j^{(3)}}{\partial W_{ji}^{(3)}} = Y_i^{(2)} \quad (2.21)$$

$$\frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}} = g'(I_j^{(3)}) \quad (2.22)$$

$$\frac{\partial E}{\partial Y_j^{(3)}} = -(d_j - Y_j^{(3)}) \quad (2.23)$$

Na qual o d_j é a saída desejada pre determindado para um sistema de aprendizagem supervisionado ou aprendizado com professor.

No entanto, a atualização dos pesos sinápticos é feita tomando o sentido contrário ao do gradiente com a finalidade de alcançar o erro mínimo.

$$\Delta W_{ji}^{(3)} = \eta * \delta_j^{(3)} * Y_i^{(2)} \quad (2.24)$$

Em que $\delta_j^{(3)}$ representa a definição do gradiente e é deduzido da seguinte forma:

Simplificando a equação 2.24 a atualização pode ser realizada iterativamente

$$\delta_j^{(3)} = (d_j - Y_j^{(3)}) * g'(I_j^{(3)}) \quad (2.25)$$

com a expressão a seguir:

$$W_{ji}^{(3)}(t + 1) = W_{ji}^{(3)}(t) + \eta * \delta_j^{(3)} * Y_j^{(2)} \quad (2.26)$$

A atualização das sinapses das camadas intermediárias são realizadas de forma distinta aos de saída da rede neural artificial, pois, elas dependem dos ajustes feitos nos neurônios que os antecedem.

Se tratando da arquitetura mostrada na figura 2.21, as respectivas camadas escondidas deveriam ser ajustadas, depois que os pesos dos neurônios da camada de saída do modelo forem atualizado.

O ajuste das sinapses $W_{ji}^{(2)}$ na segunda camada intermediária, tem o intuito de reduzir significativamente o erro correspondente a saída desta da RNA em função da anomalia do *backpropagation* proveniente da atualização da camada de saída. Com isso, segue as seguintes equações:

$$\nabla E^{(3)} = \frac{\partial E}{\partial W_{ji}^{(3)}} = \frac{\partial E}{\partial Y_j^{(3)}} * \frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}} * \frac{\partial I_j^{(3)}}{\partial W_{ji}^{(3)}} \quad (2.27)$$

Por outro lado, e com base na equação anterior, associa-se:

$$\frac{\partial I_j^{(2)}}{\partial W_{ji}^{(2)}} = Y_i^{(1)} \quad (2.28)$$

$$\frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}} = g'(I_j^2) \quad (2.29)$$

$$\frac{\partial E}{\partial Y_j^{(2)}} = \sum_{k=1}^{n_3} \frac{\partial E}{\partial I_k^{(3)}} * W_{kj}^{(3)} \quad (2.30)$$

No entanto, a atualização dos pesos sinápticos é feita tomando o sentido contrário ao do gradiente com a finalidade de alcançar o erro mínimo.

$$\Delta W_{ji}^{(2)} = \eta * \delta_j^{(2)} * Y_i^{(1)} \quad (2.31)$$

Em que δ representa a definição do gradiente e é deduzido da seguinte forma:

$$\delta_j^{(2)} = -\left(\sum_{k=1}^{n_3} \delta_{kj}^{(3)}\right) * g'(I_j^2) \quad (2.32)$$

Simplificando a equação 31 a atualização pode ser realizada iterativamente com a expressão a seguir:

$$W_{ji}^{(2)}(t+1) = W_{ji}^{(2)}(t) + \eta * \delta_j^{(2)} * Y_i^{(1)} \quad (2.33)$$

O processo da atualização dos pesos na primeira camada intermediária da figura 2.21 segue a mesma analogia o da segunda camada intermediária. O ajuste das sinapses $W_{ji}^{(1)}$ na segunda camada intermediária, tem o intuito de reduzir significativamente o erro correspondente a saída desta da RNA em função da anomalia do *backpropagation* proveniente da atualização da segunda camada escondida. Com isso, segue as seguintes equações:

- Atualização das sinapses da primeira camada intermedia

$$\nabla E^{(1)} = \frac{\partial E}{\partial W_{ji}^{(1)}} = \frac{\partial E}{\partial Y_j^{(1)}} * \frac{\partial Y_j^{(1)}}{\partial I_j^{(1)}} * \frac{\partial I_j^{(1)}}{\partial W_{ji}^{(1)}} \quad (2.34)$$

Por outro lado, e com base na equação anterior, associa-se:

$$\frac{\partial I_j^{(1)}}{\partial W_{ji}^{(1)}} = X_i \quad (2.35)$$

$$\frac{\partial Y_j^{(1)}}{\partial I_j^{(1)}} = g'(I_j^{(1)}) \quad (2.36)$$

$$\frac{\partial E}{\partial Y_j^{(1)}} = \sum_{k=1}^{n_3} \frac{\partial E}{\partial I_k^{(2)}} * W_{kj}^{(2)} \quad (2.37)$$

No entanto, a atualização dos pesos sinapcticos é feito tomando o sentido contrario ao do gradiente com a finalidade de alcançar o erro minimo.

$$\Delta W_{ji}^{(1)} = \eta * \delta_j^{(1)} * X_i \quad (2.38)$$

Em que δ representa a definição do gradiente e é deduzido da seguinte forma:

$$\delta_j^{(2)} = -\left(\sum_{k=1}^{n_2} \delta_{kj}^{(2)} * W_{kj}^{(2)}\right) * g'(I_j^{(1)}) \quad (2.39)$$

Simplificando a equação 39 a atualização pode ser realizada iterativamente com a expressão a seguir:

$$W_{ji}^{(1)}(t + 1) = W_{ji}^{(1)}(t) + \eta * \delta_j^{(1)} * X_i \quad (2.40)$$

A realização das etapas do *forward* e *backward* decorrem de forma interativa, a quantidade das interações dependem de um número pré-determinado ou até que

se chegue no número de repetições suficiente a ponto que o menor valor do erro seja alcançado com a parada antecipada, isso acontece quando o processo de interação termina antes mesmo que seja alcançado o quantidade de vezes pré-determinado, antes do número de epochs definido.

2.9.2 Desempenho de Modelos de multicamadas com Retropropagação

Apesar da constante evolução da área de machine learning a análise do desempenho dos modelos e dos algoritmos continuam sendo um enorme desafio para os estudiosos e profissionais no assunto, por esse motivo, a definição dos parâmetros tendem a ser definido de forma experimental até que se encontre o modelo adequado.

Para as redes neurais artificiais umas das funções fundamentais que devem ser escolhidos com muito cuidado são as funções de ativação, pois, é através dela que é definido o tipo de saída que deverá ser produzido pelo modelo, são divididos basicamente em dois tipos diferentes, aqueles projetados para operações lineares e outros para situações não lineares. O primeiro tem um esforço computacional ligeiramente reduzido se comparado com o segundo, porém, o segundo se adequa a quaisquer problema.

Continuando com o desempenho das RNA outro parâmetro a ser considerado é a taxa de aprendizagem, define o tempo de convergência levando em conta o tipo da taxa, ou seja, taxa fixa ou dinâmica. Tem outro parâmetro relacionado ao valor a ser atribuído a taxa de aprendizagem, que diz que se a taxa for fixa e o seu valor um tanto quanto elevado o processo de convergência tende a ser impreciso e não alcançar o mínimo local, o recomendável é que inicialmente seja atribuído valor pequeno próximo de zero e ir variando crescentemente dentro de uma taxa dinâmica pré-estabelecida até localizar um valor adequado para o hiperparâmetro da taxa de aprendizagem.

2.9.3 Treinamento com o Método *Resilient Propagation*

Em um determinado problema pode ser o caso, que seja necessário usar dois ou mais tipos de função de ativação para garantir maior confiabilidade no processo de treinamento, em modelos de perceptron multicamadas o mais indicado é a função de ativação *ReLU* nas camadas intermediárias. A função *ReLU* tem como princípio computar os valores recebidos e produzir uma saída dentro de um intervalo entre zero a infinito, para valores que entram com sinais negativos são zerados no instante da saída, esse facto faz com que treine a rede com maior celeridade, porém, tal situação leva a derivada a valores zero em consequencia disso pesos sinapticos dessa região não são atualizados.

De modo a corrigir o dilema da magnitude do gradiente, este método, limita-se apenas na correção da taxa de variação dos sinais para definir o sentido dos ajustes das sinapses, com isso, o processo de interação da taxa de aprendizagem deixa de ser estatica sendo que, no instante em que os valores do gradiente tornarem-se iguais dentro de duas interações de forma consecutiva abre a possibilidade de se elevar o valor da taxa de aprendizagem pelo fato desta estar distante do alcance do minimo local. No entanto, quando os valores do gradiente tornarem-se distintos conclui-se portanto, que deve ser diminuido o valor da aprendizagem porque o minimo local foi passado, nesse caso a redução da taxa visa melhor melhorar o processo da convergencia.

$$\Delta_{ji(L)} = \begin{cases} \eta^+ \cdot \Delta_{ji(L)}(t-1), & \text{se } \frac{\partial E(t-1)}{\partial W_{ji}^{(L)}} \cdot \frac{\partial E(t)}{\partial W_{ji}^{(L)}} > 0 \\ \eta^- \cdot \Delta_{ji(L)}(t-1), & \text{se } \frac{\partial E(t-1)}{\partial W_{ji}^{(L)}} \cdot \frac{\partial E(t)}{\partial W_{ji}^{(L)}} < 0 \\ \Delta_{ji(L)}(t-1), & \text{se } \frac{\partial E(t-1)}{\partial W_{ji}^{(L)}} \cdot \frac{\partial E(t)}{\partial W_{ji}^{(L)}} = 0 \end{cases} \quad (2.41)$$

Em que:

$\Lambda_{ji}^{(L)}$ representa a taxa de aprendizagem de forma isoladas e pertecente a somente uma sinapse, sendo que, quando $(\eta^+ > 1)$ aumenta a taxa de aprendizagem e $(0 < \eta^- < 1)$ é a redução.

3.Conclusão

Neste fase foram apresentados introdutoriamente aspetos inerentes sobre as redes neurais artificiais, primeiro passo foram contextualizados o funcionamento do cerebro biologico,

4. Aplicação das redes neurais artificiais para a localização de faltas nas linhas de transmissão de energia elétricas

A utilização de redes neurais artificiais para o setor elétrico é justificável, pois, uma das suas principais aplicações é o reconhecimento de padrões, a isso, ainda adiciona-se o facto de que elas lidam perfeitamente com problemas não lineares [(Da Silva)].

Um modelo equivalente do sistema será detalhado com a finalidade de simular os transitorios nas linhas de transmissão, a extração dos sinais a caracterização dos dados, a construção de um banco de dados e definir os parametros que nos levaram a localização dos disturbios no sistema.

4.1 Sistema simulado

Para garantir maior celeridade no processo de simulação dos curtos circuitos foi necessario o desenvolvimento de um algoritmo que rodou e simulou de forma automatica os arquivos no ATP do modelo equivalente da figura 4.1. Das simulações feitas gerou-se um banco de dados onde foram armazenados os dados para o treinamento dos modelos da rede neural artificial.

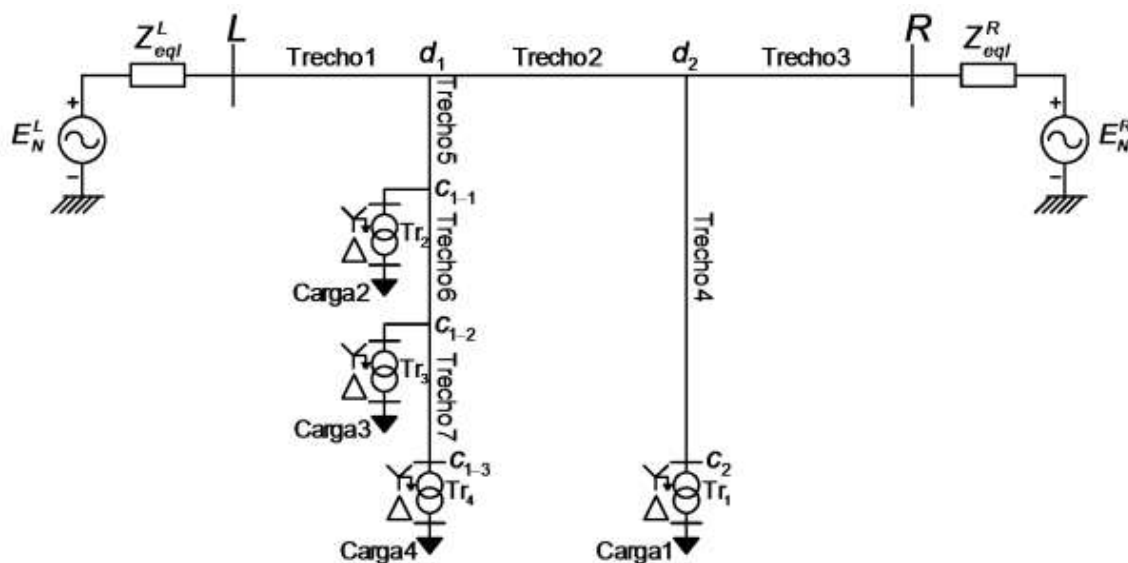


Figura 4.1 - Linha de transmissão utilizada para simulação e obtenção dos dados(Di Santo).

Em que:

E_N^L – Gerador local

E_N^R – Gerador remoto

K_{eq}^L – Impedância equivalente por trás da barra local

K_{eq}^R - Impedância equivalente por trás da barra remota

L e R – Barras local e remota

Para que o modelo de localização de defeitos funcione de forma adequada na identificação dos locais defeituosos com maior precisão, é necessário que sejam definidos certos criterios para a linha de transmissão e introduzi-los no ATP. Com tudo, os seguintes componentes foram submetidos para a simulação:

- O tipo da falta: É referente ao envolvimento das fases defeituosas e a terra. Para essa dissertação, as faltas abrangidas são: AG, BG, CG, AB, BC, AC, ABG, BCG, ACG, ABC;
- Localização da falta: São mensurados desde o ponto de obteção dos sinais até proximo do final de cada trecho ou seja, foram considerados de 0,2 a 98% de cada trecho da linha, sendo 0.2% proximo do início de cada trecho e 98% proximo do final de cada trecho;
- Impendância de falta: Acontece entre os condutores de fase e fase, ou com envolvimento do condutor terra, e os valores foram variados dentro do intervalo entre o 0,1; 1, 5; 25; 50; 100; 250; 500 ohms;
- Ângulo falta: A definição do ângulo tem muito haver com a natureza do gerador e dos sistemas a ele conectados, outro parametro que não deve ser ignorado é o sistema de aterramento[2], a seguir fazer-se-a um esclarecimento sobre esse processo. No instante em que uma tensão CA é aplicada um siatema e os valores atribuidos a R e L são constante:

$$V = \text{sen}(\omega t + \alpha) = Ri + L \frac{di}{dt} \quad (4.1)$$

Com base na equação 4.2, Inicialmente $t=0$ bem no instante da aplicação da tensão, portanto, o α é que definirá o modulo da tensão instantânea.

Solucionando a equação 4.2:

$$i = \frac{V}{|Z|} [\text{sen}(\omega t + \alpha - \theta) - \varepsilon^{-Ri/L} \text{sen}(\alpha - \theta)] \quad (4.2)$$

Em que:

$$|Z| = \sqrt{R^2 + (\omega L)^2} \quad (4.3)$$

E

$$\theta = \text{tg}^{-1}(\omega L/R) \quad (4.4)$$

Da equação 4.3, o primeiro termo é variado senoidalmente em função do tempo, assim sendo, ele representa o valor no estado permanente da corrente em um circuito RL em função de uma tensão aplicada. A outra parte não é periodico, e representa a componente continua da corrente do sistema. Quando o valor do conjunto permanente é diferente de zero em $t=0$, então, aparece a componente continua com a finalidade de corresponder a condição fisica que é anular-se a corrente quando o $t=0$. Por outro lado, sempre que $(\alpha - \theta) = 0$ ou $(\alpha - \theta)$ há ausencia da componente de corrente continua, no entanto, quando a falha ocorre no momento em que $(\alpha - \theta) = \pm\pi/2$, inicialmente o valor de pico da componente da corrente continua igual ao pico na componente senoidal(Stevenson Jr). Especificamente para essa dissertação o ângulo que incide nas falhas foi de 0 e 20 graus.

Agora que as informações relacionadas aos parâmetros que compõem o processo de localização de defeitos foram definidas no parágrafo anterior, chegou o momento de dar corpo ao banco de dados. Para isso, é necessário o uso de recursos computacionais ATP para simulações, tendo em vista as movimentações da tensão e corrente, com uma frequência de amostragem de 960Hz.

Levando em conta a complexidade para alimentar uma base de dados, foi necessário o desenvolvimento de um algoritmo no software MATLAB (MATHWORKS, 2021), para agilizar o processo de leitura e extração dos arquivos em EMTP-ATP, de forma a automatizar a geração e a alimentação de dados na base de dados, que servirão como entrada da rede neural artificial. É importante salientar que os sinais de corrente e tensão são armazenados na base de dados na forma fasorial, com componentes reais e imaginárias.

O desenvolvimento completo do sistema foi feito usando vários algoritmos, mas em apenas dois tipos de software. Para as simulações e a obtenção dos dados que alimentaram o banco de dados, o algoritmo foi desenvolvido no MATLAB. Já para os modelos de redes neurais artificiais, foi utilizado o PYTHON, com as suas *frameworks*, como o SCIKIT-LEARN e TENSORFLOW. Para mais detalhes sobre as funcionalidades de cada um dos softwares, serão abordados nos próximos capítulos.

4.2 Detecção da falta

A detecção de defeitos nas linhas de transmissão é fundamental para garantir a qualidade do funcionamento do sistema. Para isso, sinais de tensão e corrente dos geradores são coletados e passam por um filtro passa-baixa de Butterworth. Os sinais de tensão e corrente são, então, reamostrados em (64) amostras cada um. O objetivo da aplicação do filtro é evitar o aliasing nos sinais de tensão e corrente, antes de serem armazenados na base de dados, já que os modelos de redes neurais são sensíveis a variáveis fora dos padrões ou outlier. O dispositivo de proteção atuará em 7ms depois do surgimento do defeito.

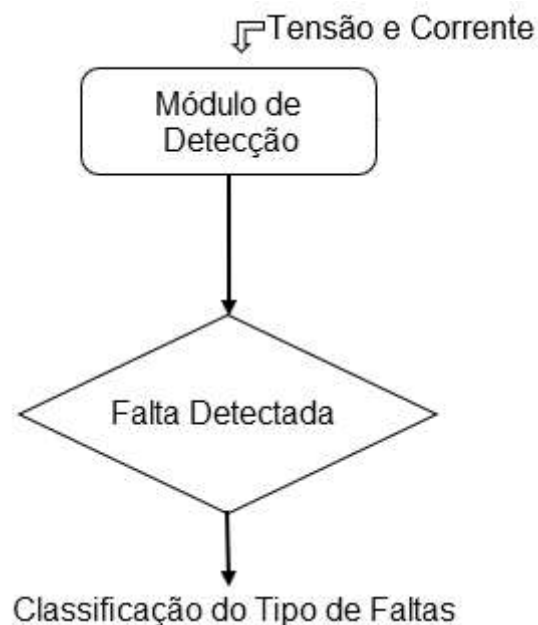


Figura 4.2 – Módulo de detecção de faltas

A figura 4.2 mostra o processo de detecção de faltas na linha de transmissão, que é crucial para garantir a qualidade de funcionamento do sistema. Os sinais de tensão e corrente são coletados de dois geradores com uma frequência de 940 Hz e passam por um filtro passa-baixa butterworth para evitar o aliasing. Em seguida, os sinais são reamostrados em 64 amostras para cada sinal.

Depois, os sinais pré-faltas e pós-falta das tensões e correntes são obtidos para melhorar a eficiência e evitar dados desnecessários. A base de dados armazena somente os sinais de tensão e corrente com faltas, calculados como a diferença entre os sinais fasoriais pós-falta e pré-falta. Com esse processo, a base de dados foi exugada e teve sua qualidade aprimorada, com 93816 pontos que alimentaram os modelos na fase de treinamento da rede neural artificial.

A figura 4.3 representa as 24 entradas para a rede neural artificial, onde cada entrada é composta pelos valores fasoriais da tensão e corrente, que são reamostrados a uma taxa de 940 Hz. A reamostragem permite que os padrões de tensão e corrente sejam corretamente estimulados na entrada da rede neural, garantindo uma classificação mais precisa dos sinais de entrada. É importante destacar que a quantidade de entradas para o modelo foi determinada após análises detalhadas de eficiência dos modelos, levando em consideração e equilíbrio entre precisão e eficiência computacional.

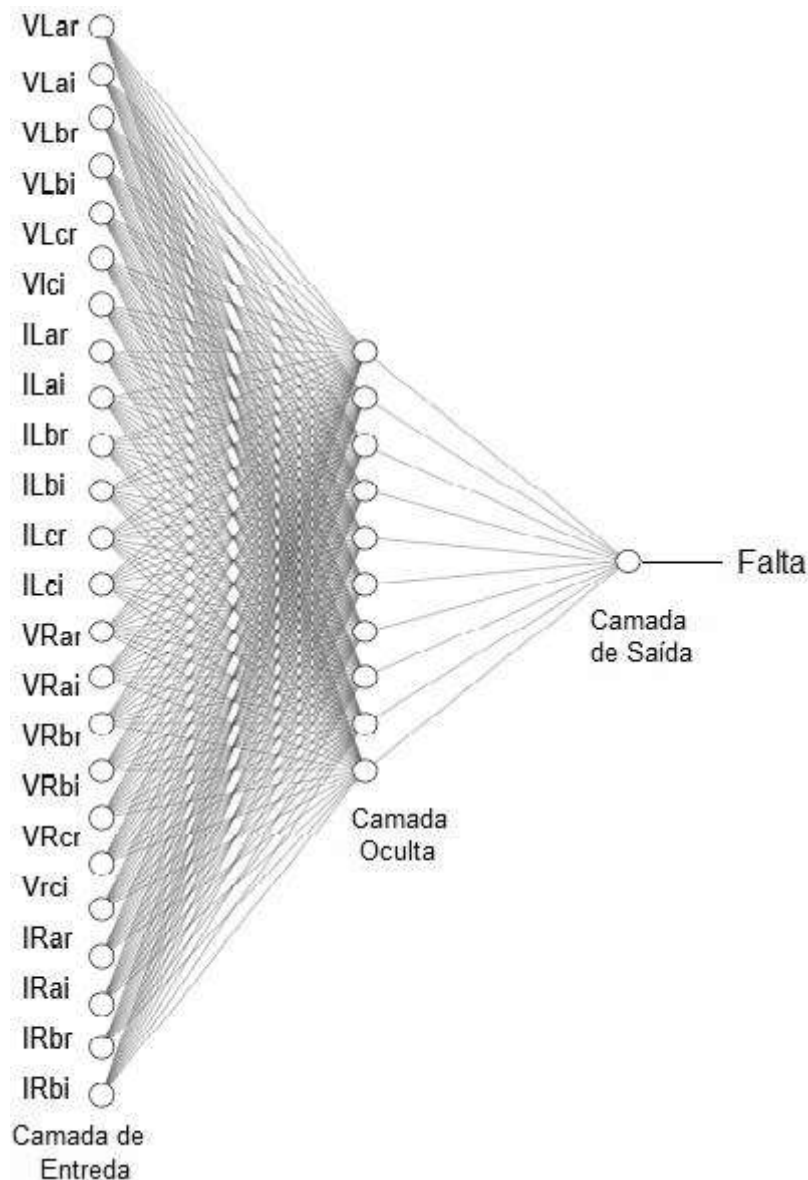


Figura 4.3 – Rede Neural com 24-10-1 para detecção

O modelo de RNA é composta por uma camada intermediária com 10 neurônios e uma camada de saída. A arquitetura utilizada é a de rede perceptron multicamadas, do tipo de aprendizagem supervisionada, e a *feedforward*. A padronização dos sinais de entrada foi feita na escala entre -1 e 1, pois esses modelos são muito sensíveis às variações nos valores das variáveis. A padronização é importante porque os algoritmos de *machine learning* são bastantes sensíveis a discrepância nos valores das suas variáveis.

Em relação ao modelo da RNA que representa o pedaço da detecção figura 4.3, ela é constituída por uma camada intermediária e composta por 10 neurónios e 1 camada de saída. Do tipo de aprendizagem supervisionada e uma rede *perceptron* multicamadas da arquitetura *feedforward*, no que se refere a padronização dos sinais de entradas, foram padronizados dentro da escala entre (-1, 1) é necessário padronizar, pois os algoritmos de *machine learning* são bastantes sensíveis a discrepância nos valores das suas variáveis.

Após a padronização dos sinais de entradas, foi então, também feita a para os sinais alvo que nesse caso foram padronizados no intervalo de (0 e 1).

Para a camada escondida foi usado como função de ativação o do tipo *ReLU* (*Rectified Linear Unit*) ela é amplamente utilizada devido a sua eficiência na convergência. E na saída da rede foi utilizado a função do tipo linear que apesar das suas limitações a função de ativação linear é apropriada para situações linearmente separáveis, o que é desejável para o problema de detecção.

Quanto ao treinamento foi usado o método *resilient-propagation* que é uma versão melhorada do *backpropagation*, onde os dados foram divididos em duas partes distintas e ficou assim: 70% para treinamento e 30% dos dados para o teste do modelo, ainda sobre o teste é fundamental que eles fiquem inacessíveis durante o processo de treinamento, pois, a fase do teste é que representam o funcionamento do modelo no mundo real.

Para a detecção foi escolhido o método de validação holdout, que consiste em separar parte dos dados para treinamento e a outra parte para o teste, o treinamento convergiu de forma antecipada.

4.3 Classificação do tipo de faltas

Agora com os defeitos detectados, os sinais são enviados para o módulo do conjunto de classificação, que tem como premissa analisar os tipos de faltas envolvidos no sistema da linha de transmissão, ou seja, quando uma falta é detecta o módulo de classificação entra em ação detectando os sinais envolvidos, como mostra a figura 4.4.

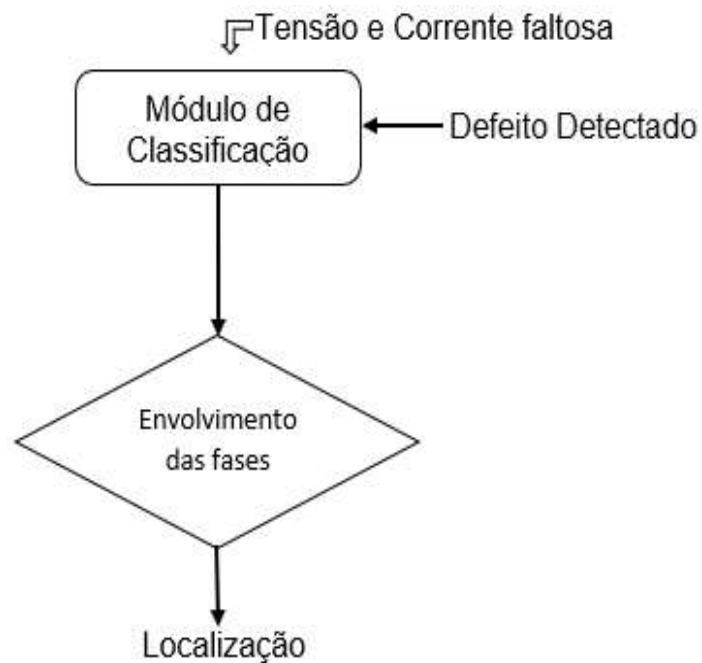


Figura 4.4 – Módulo de classificação do tipo de falta

Esse módulo tem única e exclusiva função de classificar o envolvimento do tipo de fases na ocorrência de alguma anomalia no sistema, a arquitetura do modelo do tipo de faltas é apresentada na figura 4.5.

O modelo da RNA da figura 4.5, apresenta uma arquitetura diferente da do modelo de detecção, principalmente nas camadas intermediárias e na de saída. Nesta, existem 2 camadas intermediárias e cada camada possui 13 neurónios agrupados verticalmente, 1 camada de saída com 4 neurónios que representam as fases A, B, C e incluindo a fase terre G.

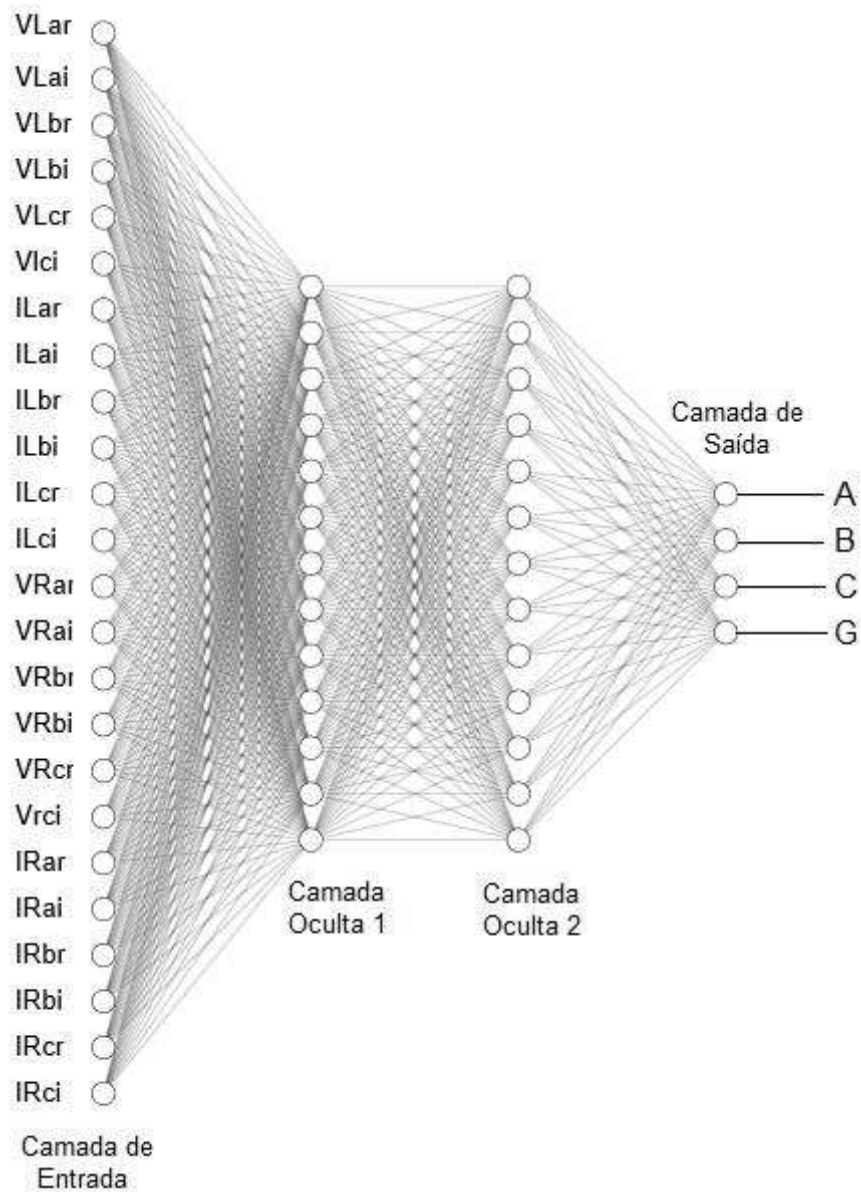


Figura 4.5 – Rede Neural com 24-13-13-4 para classificação de faltas

Quanto a padronização dos sinais de entradas, segue o definido no modelo da detecção.

Tabela 4.1 – Tabela da verdade classificação de faltas

Tipo de faltas	Saídas			
	A	B	C	G
AG	1	0	0	1
BG	0	1	0	1
CG	0	0	1	1
ABG	1	1	0	1
BCG	0	1	1	1
CAG	1	0	1	1
AB	1	0	0	0
BC	0	1	1	0
CA	1	0	1	0
ABC	1	1	1	0

A variável alvo que representam as saídas da arquitetura da figura 4.5 foram normalizados entre (0, 1), onde, o 0 remete-nos a ausência de envolvimento de fases defeituosa e o 1 significa que foi classificado um determinado tipo de falta, vide a tabela 4.1

Quanto ao treinamento foi usado o método *resilient-propagation* que é uma versão melhorada do *backpropagation*, como no modelo anterior e tipo empregado no processo de aprendizado é o supervisionada, ou seja, aprende com um professor, com a mesma técnica usada no modelo da detecção, o mesmo se repetiu com o uso de um *perceptron* multicamadas da arquitetura *feedforward*.

Para a função de ativação foram empregues dois tipos diferentes, para as duas camadas escondidas foi usado a função de ativação o do tipo *ReLU*, e na saída da rede neural artificial foi do tipo sigmoide, diferente da função linear é adequado para problemas não lineares.

Tabela 2 – Erro na classificação

Tipo de faltas reais por trechos	Respostas da rede neural							
	Trec1	Trec2	Trec3	Trec4	Trec5	Trec6	Trec7	Erro
AG	AG	AG	AG	AG	AG	AG	AG	0,00
BG	BC	BC	BC	BC	BC	BC	BC	0,00
CG	CG	CG	CG	CG	CG	CG	CG	0,00
ABG	ABG	ABG	ABG	ABG	ABG	ABG	ABG	0,00
BCG	BCG	BCG	BCG	BCG	BCG	BCG	BCG	0,00
CAG	CAG	CAG	CAG	CAG	CAG	CAG	CAG	0,00
AB	AB	AB	AB	AB	AB	AB	AB	0,00
BC	BC	BC	BC	BC	BC	BC	BC	0,00
CA	CA	CA	CA	CA	CA	CA	CA	0,00
ABC	ABC	ABC	ABC	ABC	ABC	ABC	ABC	0,00

Trec = Trecho

4.4 Localização de faltas

Finalmente chegou a etapa da localização dos pontos com defeitos da linha de transmissão, como mostrado na figura 4.6

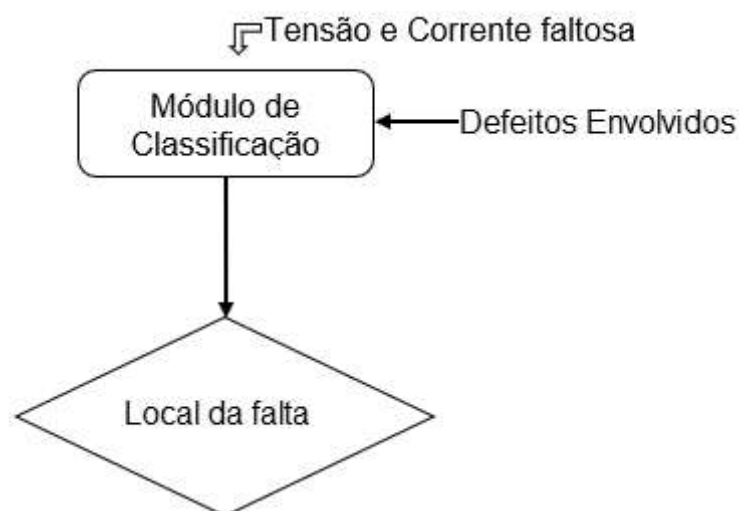


Figura 4.6 – Localização de faltas

Visto que, o processo de classificação do tipo de falta foi concluído, então, valores de faltas de meio ciclo são apresentados ao conjunto que define o local da ocorrência da falta, na posse das informações sobre as faltas envolvidas cabe então determinar modelos específicos para cada tipo de faltas, como mostra a arquitetura básica do modelo de localização de faltas na figura 4.7.

A composição dos parâmetros do modelo para a localização de faltas são ligeiramente complexa se comparado com as outras duas anteriores, sendo que, a variável a ser alcançada apresenta uma ordem bastante variada, e isso faz com que seja solicitado maior gastos computacionais e software com maior eficiência para redes neuais.

No que concerne a variável alvo tem os mesmos criterios adotados na classificação do tipo de faltas em que as saídas da arquitetura da figura 4.7 foram normalizados entre (0, 1), onde, o 0 indica que nenhuma defeito foilocalizado ao longo dos trechos da linha de transmissão, já o número 1 significa que foi localizado uma determinada falta em dos trechos.

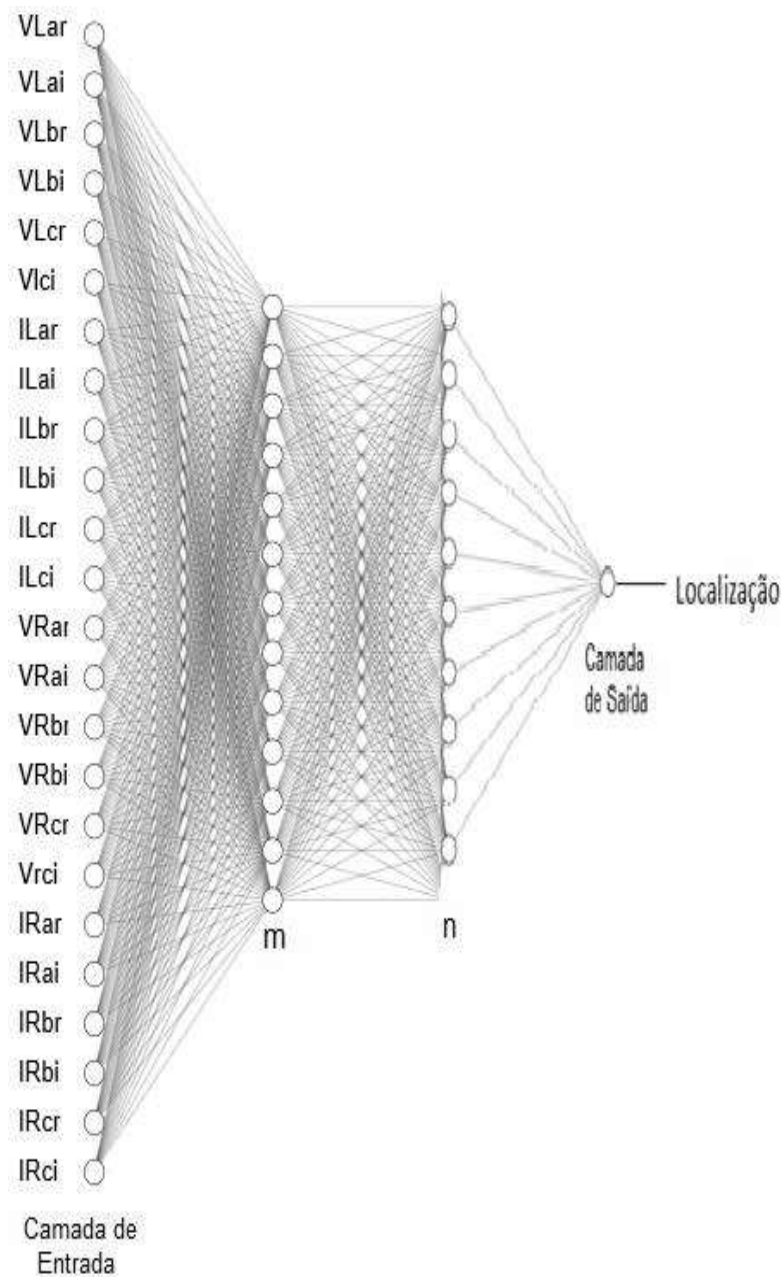


Figura 4.7 – Rede Neural com 24 – m - n -1 para localização de faltas

Para a função de ativação foram empregues dois tipos diferentes, para as duas camadas escondidas foi usado a função de ativação o do tipo Relu, e na saída da rede neural artificial foi do tipo sigmoide, diferente da função linear é adequado para problemas não lineares.

Quanto ao treinamento foi usado o método *resilient-propagation* que é uma versão melhorada do *backpropagation*, como no modelo anterior e tipo

empregado no processo de aprendizado é o supervisionada, ou seja, aprende com um professor, com a mesma técnica usada no modelo da detecção, o mesmo se repetiu com o uso de um *perceptron* multicamadas da arquitetura *feedforward*.

5. Conclusão

Nessa etapa, as estruturas do conjunto completo que compõem as três fases dos sistemas de localização de curto-circuito posto neste trabalho foram descritas de forma detalhada.

Outro parametro descrito foi o modelo da linha de transmissão estudado, que foi simulado através de um código desenvolvido no MATLAB variando os diferentes tipos de faltas assim como suas respectivas impedâncias. Ainda foram definidas as componentes que formam as topologias dos modelos das RNA.

6.Topologia da Rede Neural Artificial

Para que um modelo de rede rede neural artificial alcance sucesso, é necessário definir vários parâmetros, hiperparâmetros, métricas e outros componentes, que serão apresentados nos paragrafos subsequentes, para garantir o melhor desempenho durante o treinamento.

Especificamente para este trabalho, a métrica adotada para medir a qualidade e a eficacia dos resultados dos algoritmos dos modelos e definir a arquitetura da rede neural artificial é o *mean squared error* Erro Médio Quadrático) ou *Root Mean Squared Error* (Raiz Quadrada do Erro Médio Quadrático). Assim, o modelo que apresentar menor erro possível será o escolhido.

Em que:

$$MSE = \sum_{i=1}^n \frac{|D_i - S_i|^2}{n} \quad (6.1)$$

Ou

$$RMSE = \sqrt{\sum_{i=1}^n \frac{|D_i - S_i|^2}{n}} \quad (6.2)$$

D_i = Saída conhecida ou valor alvo

S_i = Saída predita pela rede neural artificial

n = O numero total amostrado

Os outros componentes que completam a arquitetura foram apresentados no capitulo 4 desta dissertação. Os sinais de entradas são os mesmos para todas

arquitecturas ou seja, para a arquitetura da deteção, classificação e localização como mostrado nas figuras 4.3, 4.5 e 4.7. O mesmo se sucede com o tipo de padronização ou normalização, todos os valores de entrada foram normalizados na faixa de (-1, 1)

Para a função de ativação foi definida como sendo o do tipo de ativação *ReLU* para todas as camadas escondidas de todas as arquiteturas dos modelos desenvolvidos.

Em relação a função de ativação nas camadas de saídas foram variadas mediante as situações apresentadas, para a deteção foi utilizado o tipo de função de ativação linear e para as camadas de saídas da arquitetura de classificação e de localização foram empregues o sigmoide. E o tipo de aprendizagem é supervisionado para todos, igualmente o *resilient-propagation* foi definido como o método de treinamento como detalhado no capítulo anterior.

O processo de convergência do treinamento foi definido como parada antecipada, nesse caso o processo termina antes do número de épocas definido ou então, quando o menor erro médio for alcançado.

Os componentes definidos neste capítulo são de abrangência geral, ou seja, é constante para todas as arquiteturas. Os parâmetros específicos de cada modelo serão adicionados nos passos a seguir.

6.1 Modelo de Deteção da faltas

A base de dados para a deteção recebeu 93816 sinais de amostras de tensão e corrente, desses, 70% foi reservado como variável de entrada para treinamento e os restantes 30% para o teste do desempenho do modelo, os parâmetros adicionais que completam a arquitetura deste modelo estão descritos na figura 4.3. Após o treinamento, o gráfico com o desempenho é mostrado na figura 6.1.

Depois de inúmeras tentativas para encontrar a configuração e arquitetura que apresente o melhor desempenho, foi identificado que o modelo com o menor *mean squared error* é constituído por uma camada intermediária composta com

10 neurónios, e 1 neurônio na camada de saída. Essa configuração é demonstrada na figura 4.3.

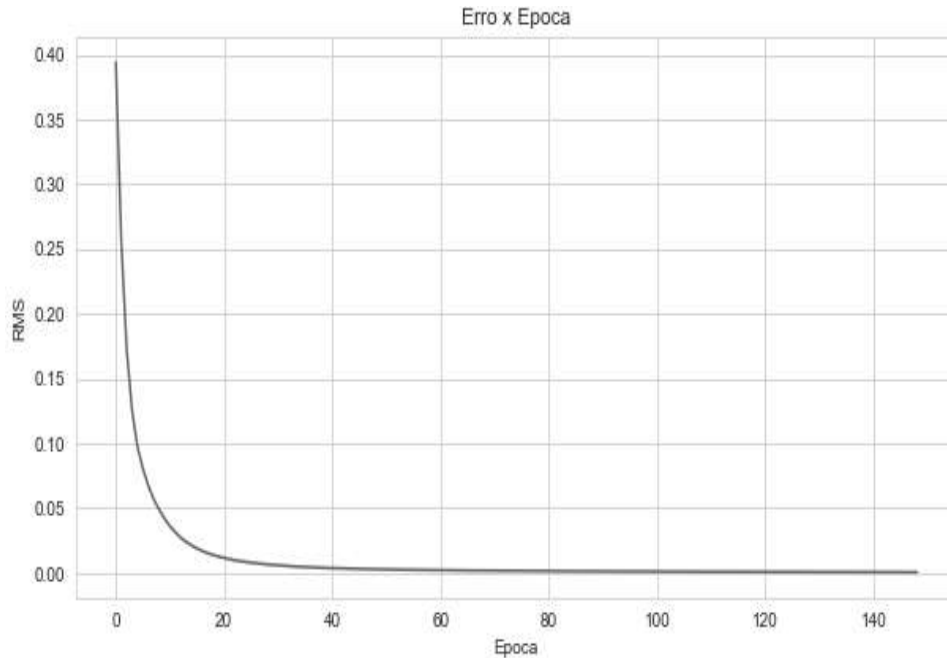


Figura 6.1 – Desempenho do módulo da detecção

Com todos os parâmetros definidos no paragrafo anterior, foi possível utilizar os 30% dos dados que ainda não haviam utilizados na fase de treino, foram usados para testar a eficácia do modelo da rede neural.

Como é visível no gráfico da figura 6.1, o processo iterativo terminou após a conclusão de 150 épocas, muito antes do número de épocas definido, que era de 1500 épocas. Isso corrobora com a tese da convergência antecipada mencionada anteriormente, tanto para o treinamento quanto para o teste. Ao analisar a trajetória do *mean squared error* em relação ao número de épocas, é possível constatar que o modelo teve 100% de acerto, visto que o erro foi igual a zero. A taxa de acertos é representada na tabela 6.1.

Tabela 6.1 – taxa de acertos

Dados	Faltas detectadas
Treinamento	100%
Teste	100%

Observando a tabela 6.1 é notório o quão as faltas foram detectadas com eficacia de 100% em um periodo de 0.05 milissendos (ms).

6.2 Modelo de Classificação da faltas

Diferentemente do modelo da detecção, em que foram utilizados todos os dados do banco de dados, no modelo de classificação foram utilizados apenas as amostras faltosas da tensão e corrente, talizando 3913 amostras. A separação dos dados entre as partes para treinamento e teste seguiu os mesmos critérios adotados no modelo de detecção.

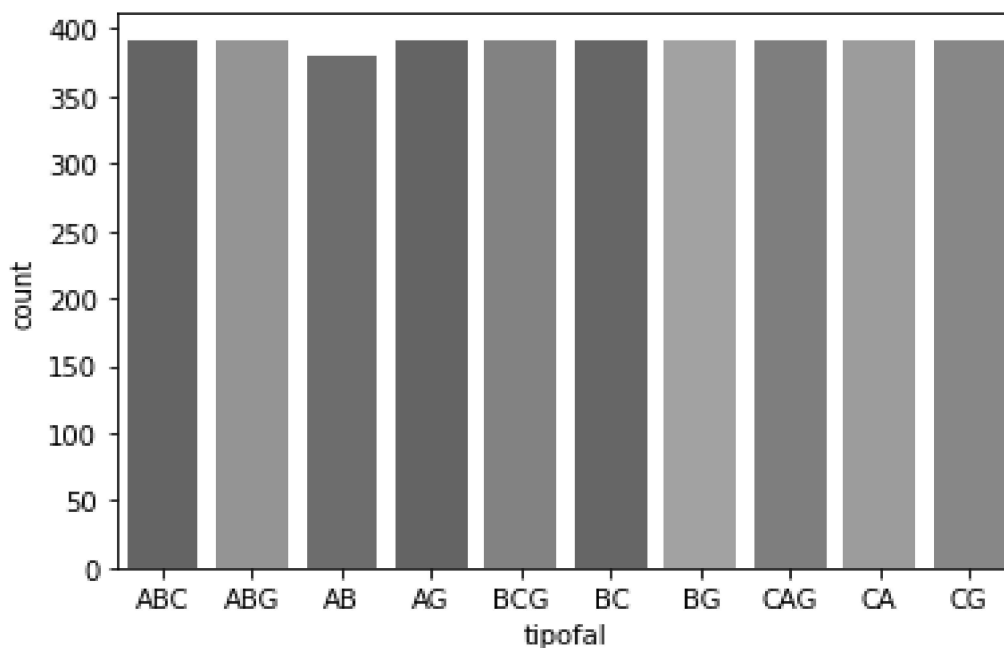


Figura 6.2 – Distribuição dos tipo de faltas

No gráfico da figura 6.2, está representada a distribuição dos dados faltosos do tipo de faltas contidos na base de dados usada como sinal de referência no paradigma de aprendizagem supervisionada.

O critério de avaliação de desempenho é mesmo utilizado no subcapítulo 5.1 com o objetivo de encontrar o menor erro médio possível, ou seja, aquele mais próximo de zero, conforme representado no gráfico da figura 5.3.

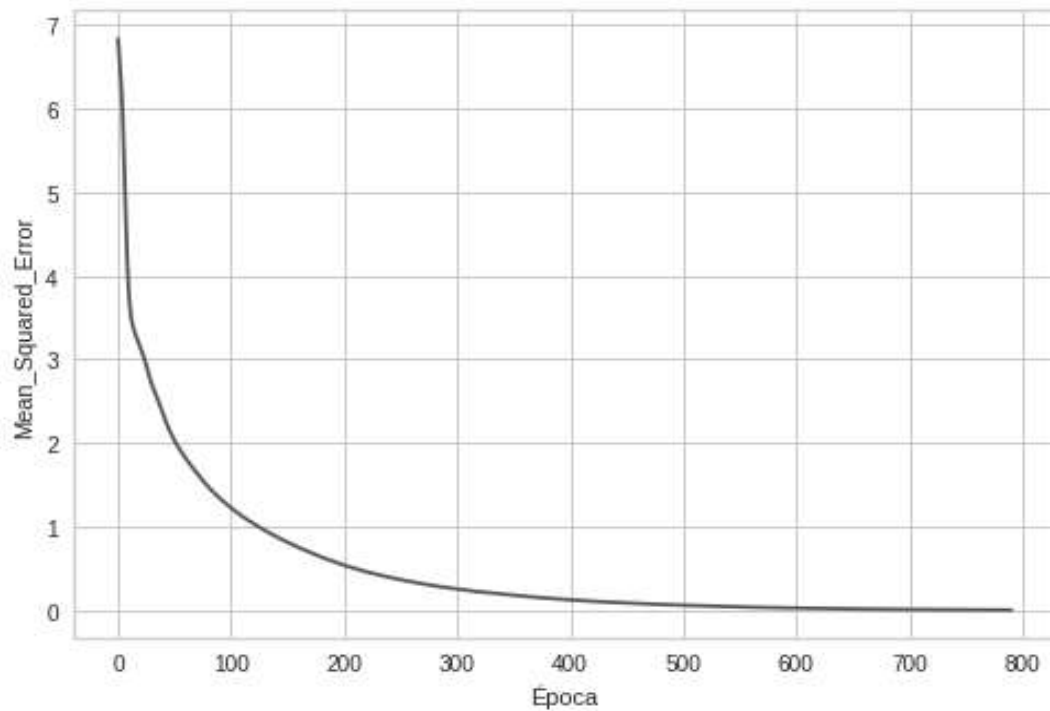


Figura 6.3 – Desempenho do módulo da classificação

No que diz respeito à escolha dos componentes que compõem os parâmetros para a configuração da arquitetura da rede de classificação de faltas, foram adotados critérios semelhantes aos demais componentes. Parte desses critérios foram especificados com detalhes no capítulo 5.

Entretanto, é importante lembrar que as camadas intermediárias, assim como seus respectivos neurônios, do modelo de classificação foram ajustados pelo método de tentativa e erro até chegar na arquitetura da figura 4.5, que apresentou o melhor desempenho com duas camadas intermediárias, cada uma composta por 13 neurônios, respectivamente. Quanto à camada de saída, ela é diferente

da do modelo de detecção, que apresenta apenas um único neurônio na camada, enquanto que no modelo de classificação a camada de saída apresenta 4 neuronios representando as fases A, B, C e G.

Tabela 6.2 – taxa de acertos da classificação

Dados	Faltas detectadas
Treinamento	100%
Teste	100%

6.3 Modelo de Localização de faltas

Antes de implementar o modelo da rede neural artificial para a localização de faltas em uma linha de transmissão, é necessário detalhar as características da própria linha de transmissão, que possui um comprimento total de 280 km e é dividida em trechos como mostrado na figura 6.4. cada trecho é identificado por um número, sendo eles: trecho 1, trecho 2, trecho 3, trecho 4, trecho 5, trecho 6 e trecho 7, respectivamente. Além disso, é essencial entender como os trechos da linha estão conectados entre si para a implementação adequada do modelo.

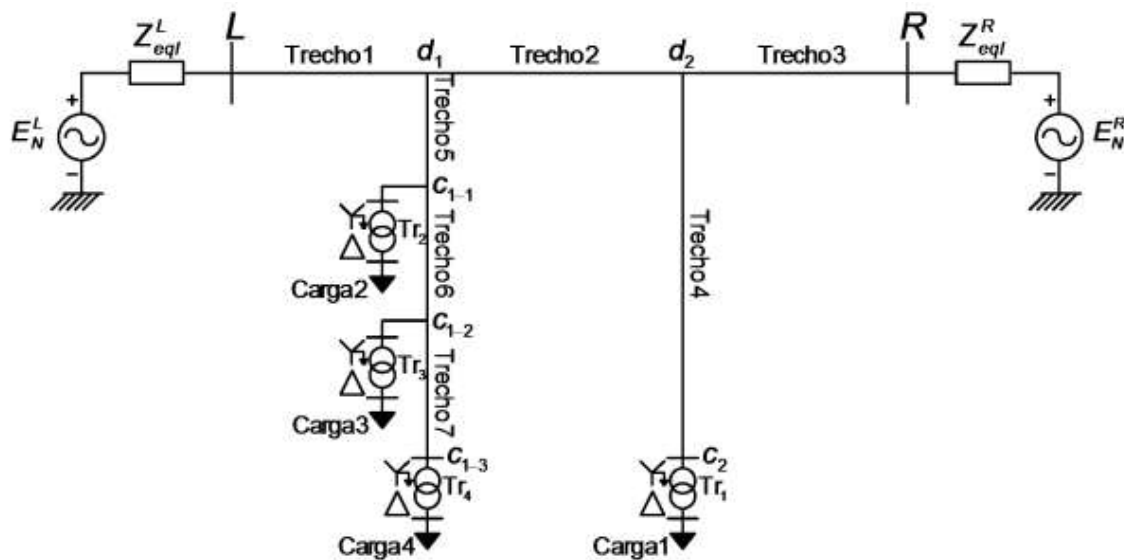


Figura 6.4 – Linha de transmissão usada para a localização de defeitos (Di Santo)

A linha de transmissão da figura 6.4 é configurado com as seguintes informações:

Tabela 6.3 – Informações da linha de transmissão

Tensão nominal	138 kV
Tipo de circuito	Simplex

6.3.1 Sistemas equivalentes dos geradores

Tabela 6.4 – características dos geradores

Tensão no gerador local	$138 \angle 0^\circ \text{kV}$
Tensão no gerador remoto	$138 \angle 20^\circ \text{kV}$

Tabela 6.5 – Sequência positiva

Terminal local		Terminal remoto	
Potência de curto circuito trifásico	$0,97 \angle 84,14^\circ \text{ GVA}$	Potência de curto circuito trifásico	$2,23 \angle 83,29^\circ \text{ GVA}$
Potência de curto circuito fase-terra	$0,72 \angle 83,46^\circ \text{ GVA}$	Potência de curto circuito fase-terra	$1,60 \angle 83,57^\circ \text{ GVA}$
Impedância equivalente	$2 + j19,5 \Omega$	Impedância equivalente	$1 + j8,5 \Omega$

Tabela 6.6 – Sequência zero

Terminal local		Terminal remoto	
Impedância equivalente	$5 + j39,5 \Omega$	Impedância equivalente	$2 + j18,5 \Omega$

7. Apresentação Dos Resultados Variando as distâncias das faltas

Para dimensionar as distâncias, foi utilizado como base em (Di Santo) que definiu o seguinte: as distâncias das faltas nos trechos 1, 2 e 3 são mensuradas tendo como referência de partida o gerador local, enquanto que, para o trecho 4, a referência é a conexão 2. Já as distâncias para os trechos 5, 6 e 7 têm como ponto de partida na conexão 1.

7.1 Parâmetros dos modelos

O processo de treinamento dos modelos das redes neurais artificiais foi realizado usando o método aprendizado supervisionado, utilizando os algoritmos implementados em python, com as bibliotecas Tensorflow e Keras. O treinamento é finalizado quando o valor do erro quadrático médio (*mean squared error*) atinge o menor valor possível, ou quando o número de épocas definido é alcançado, ou ainda através de uma parada antecipada. No caso dos modelos de localização, todos convergiram por parada antecipada.

No que diz respeito ao treinamento, foi feita uma filtragem dos dados por trecho na base de dados. Assim, o treinamento também foi realizado por trecho, variando as distâncias dentro de cada trecho, de acordo com as características do sistema da linha de transmissão. Isso significa que somente os dados relativos a cada trecho foram utilizados no treinamento correspondente, garantindo, dessa forma, maior confiabilidade nos resultados. Essa abordagem também evita vazamento de dados entre os trechos, o que pode prejudicar o desempenho do modelo.

A divisão dos dados para treinamento e teste seguiu um parâmetro diferente dos demais devido às características específicas da linha de transmissão. Os trechos com distâncias menores possuem menor quantidade de dados, enquanto os trechos mais longos têm um grande volume de dados. Por isso, os trechos 1 e 5 tiveram seus dados divididos da seguinte forma: 50% para treinamento e 50% para testes. Para o restante dos trechos, a divisão foi 70% para treinamento e 30% para o testes.

Para determinar a quantidade de camadas intermediárias e seus respectivos neurônios, utilizou-se o método de alteração dos neurônios dos modelos das redes neurais artificiais, que consiste no processo de tentativa e erro. Para a localização de faltas do tipo fase terra e do tipo fase – fase – fase - terra, alguns parâmetros de suas arquiteturas ficaram assim definidos:

Todas as arquiteturas convergiram com 2 camadas intermediárias, conforme apresentado nas tabelas 7.1 e 7.2;

Enquanto que o número de neurônios por cada camada intermediária variou de arquitetura para arquitetura, as arquiteturas com os neurônios que convergiram também estão mostradas nas tabelas 7.1 e 7.2 respectivamente;

A camada de saída contém 1 neurônio para todas as arquiteturas;

Em relação ao número de épocas, todas convergiram antecipadamente em 5 épocas, como mostrado nas tabelas 7.1, 7.2 e nos gráficos com as performances do *mean squared error*, que serão apresentados mais adiante.

A tabela 7.1 apresenta os efeitos da variação de camadas intermediárias e seus respectivos neurônios, tanto para as faltas monofásicas quanto as faltas trifásicas ambas tiveram um comportamento similar quando é variado a quantidade de neurônios nas camadas intermediárias e o mesmo sucede com a quantidade de número de épocas.

Tabela 7.1 – Variação de camadas e neurônios intermediários nas faltas fase terra

Trecho	m	n	Épocas	convergência
1	10	10	15	Não
	13	13	15	Não
	18	18	5	Sim
2	10	10	15	Não
	16	16	15	Não
	20	20	5	Sim
3	16	16	15	Não
	18	18	15	Não
	20	20	5	Sim
4	16	16	15	Não
	18	18	15	Não
	20	20	5	Sim
5	16	16	15	Não
	18	18	15	Não
	30	30	5	Sim
6	13	13	15	Não
	16	16	15	Não
	25	25	5	Sim
7	16	16	15	Não
	18	18	15	Não
	30	30	5	Sim

m – Primeira camada escondida

n – Segunda camada escondida

As métricas em redes neurais artificiais são parâmetros que servem para avaliar o desempenho, a eficácia e na escolher a arquitetura que apresenta melhores resultados. Para o modelo de localização de faltas, optou-se por utilizar erro (*loss*) como parâmetro para avaliar o desempenho dos modelos. O erro consiste na diferença entre o valor desejado e o valor predito pelo modelo de rede neural artificial, como mostrado na equação 7.1. É importante ressaltar que essa métrica segue basicamente as mesmas equações descritas em 5.1 e 5.2.

$$erro = \frac{(D-S)}{n} \quad (7.1)$$

Em que:

D = Saída conhecida ou valor alvo

S = Saída predita pela rede neural artificial

n = O numero total amostrado

7.1.1 Falta fase – terra

Os gráficos das figuras 7.1, 7.2, 7.3, 7.4, 7.5, 7.6 e 7.7 apresentam as performances dos modelos da rede neural em relação ao número de épocas. Apesar da variação da resistência e das distâncias em cada trecho, porém, os gráficos aqui mostrados contém somente os dados com resistências de 250 ohm e com as distâncias de faltas localizadas a 50% de cada trecho da linha da transmissão. Além disso, os gráficos também mostram que todas as arquiteturas confirmaram o que já foi antecipado anteriormente, em relação a antecipação da convergência no processo de treinamento e teste.

Resistencia de falta 250 ohm

Distância = 50% de cada trecho

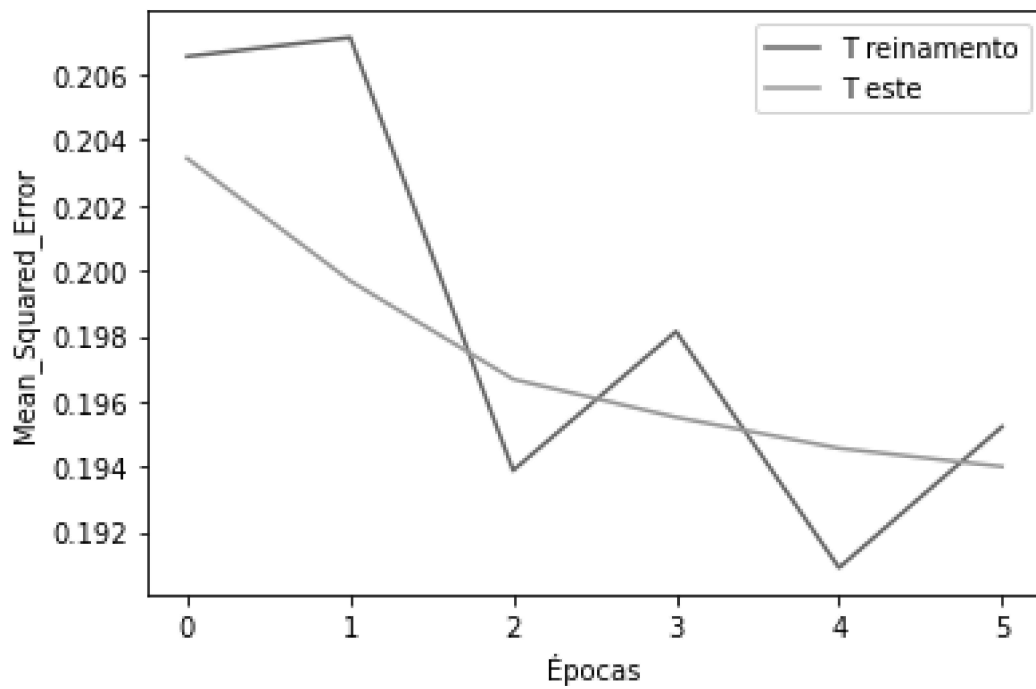


Figura 7.1 desempenho do modelo falta fase-terra 1º trecho

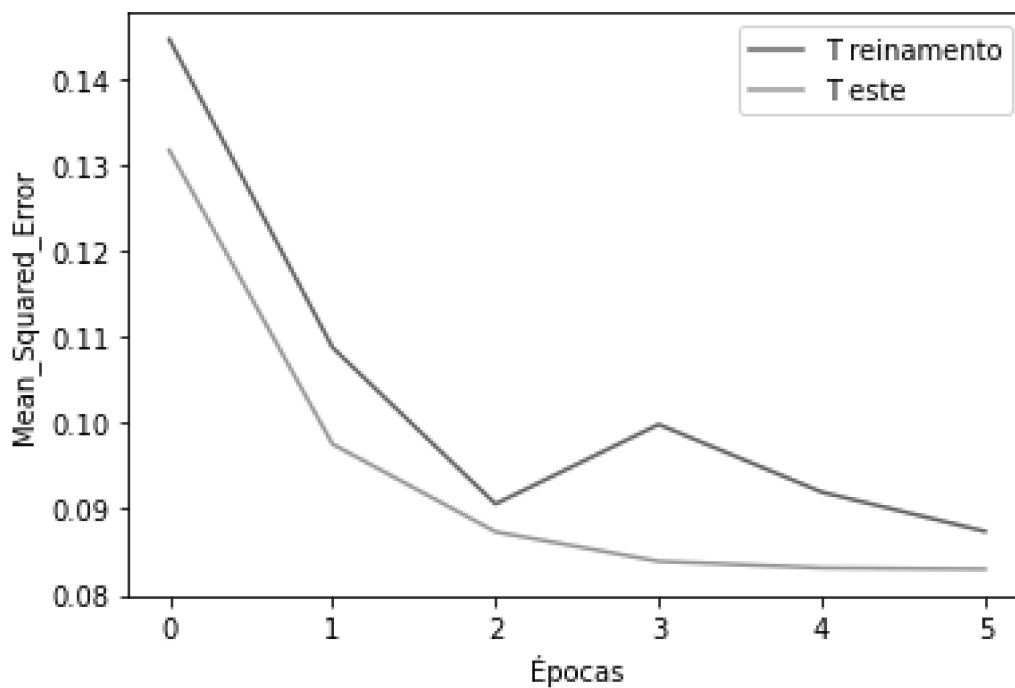


Figura 7.2 desempenho do modelo falta fase-terra 2º trecho

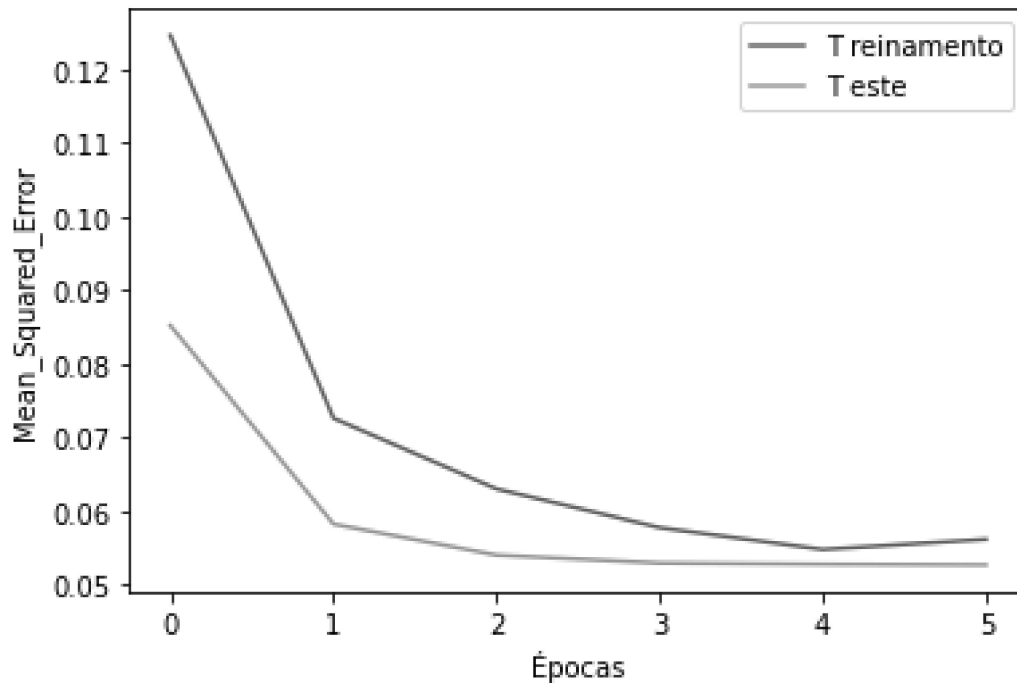


Figura 7.3 desempenho do modelo falta fase-terra 3º trecho

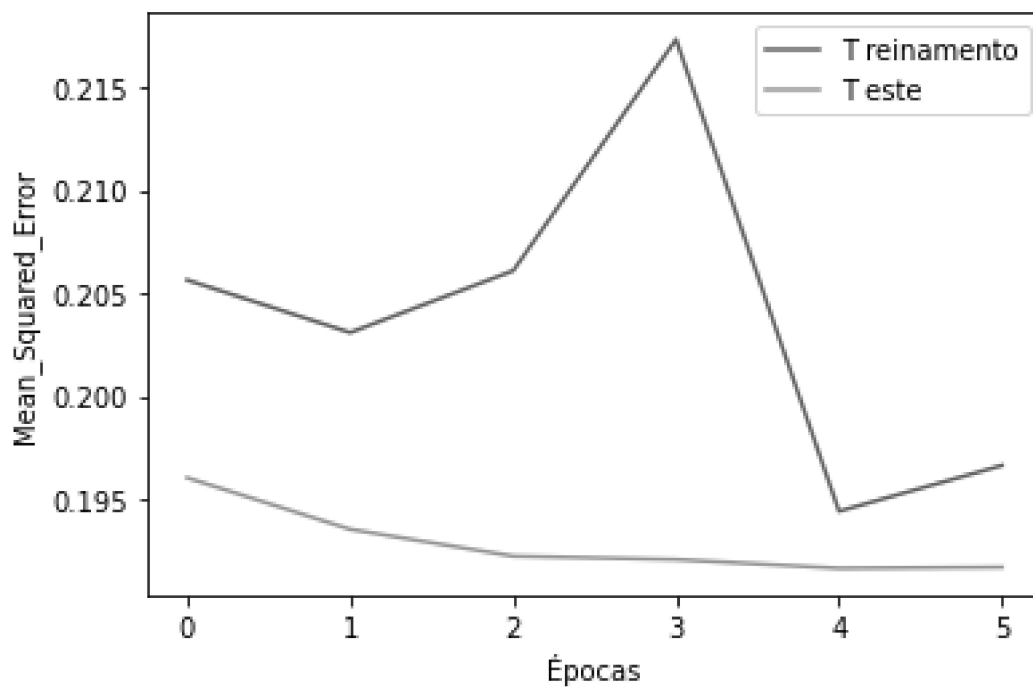


Figura 7.4 desempenho do modelo falta fase-terra 4º trecho

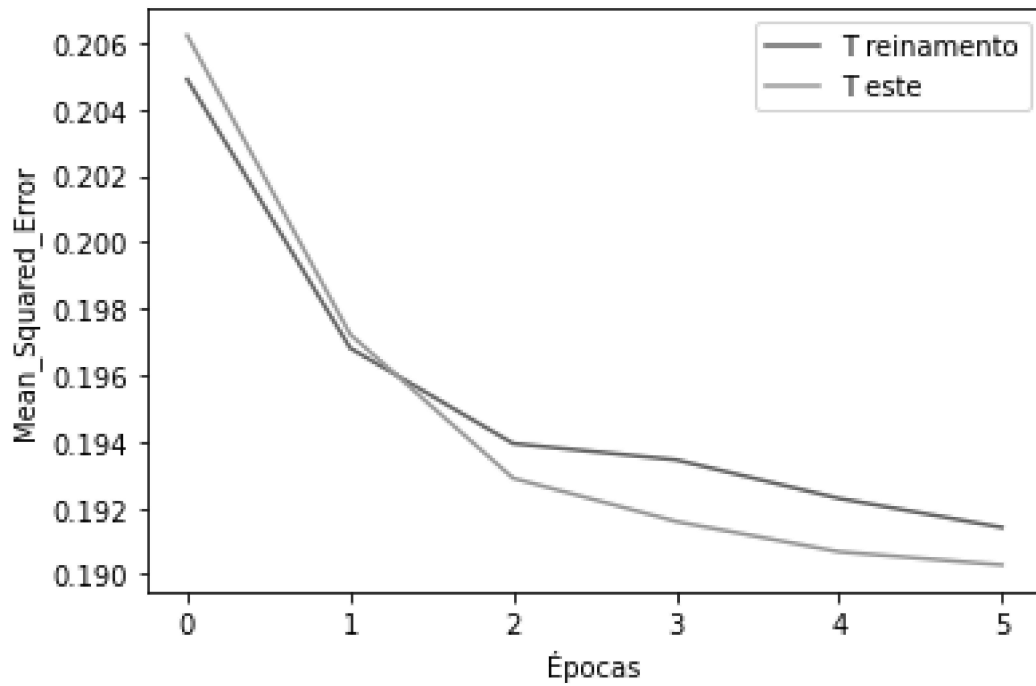


Figura 7.5 desempenho do modelo falta fase-terra 5º trecho

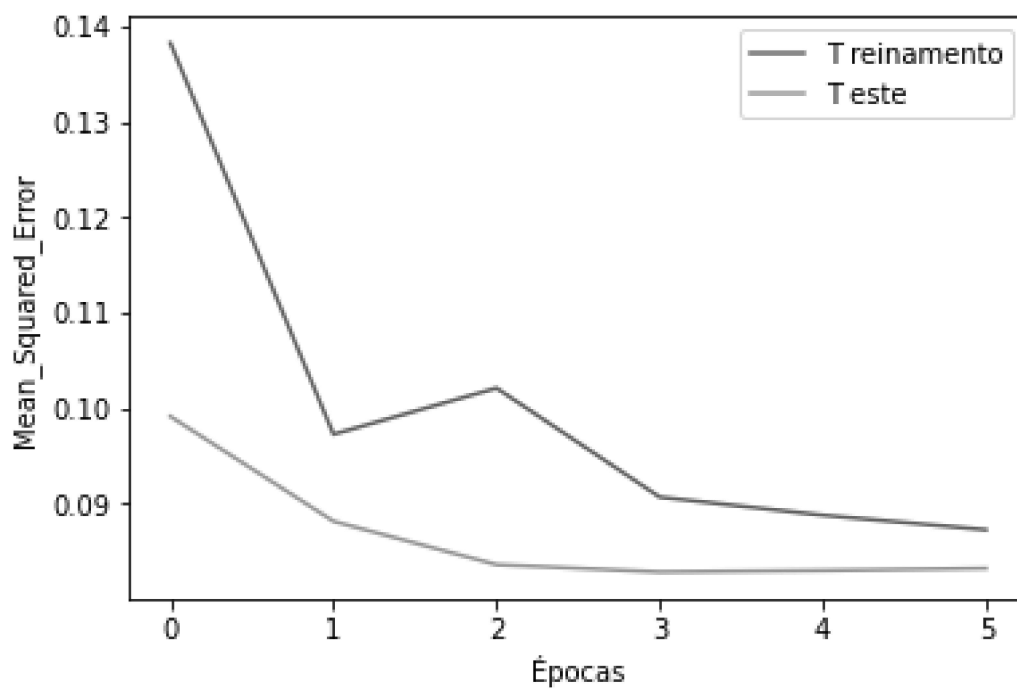


Figura 7.6 desempenho do modelo falta fase-terra 6º trecho

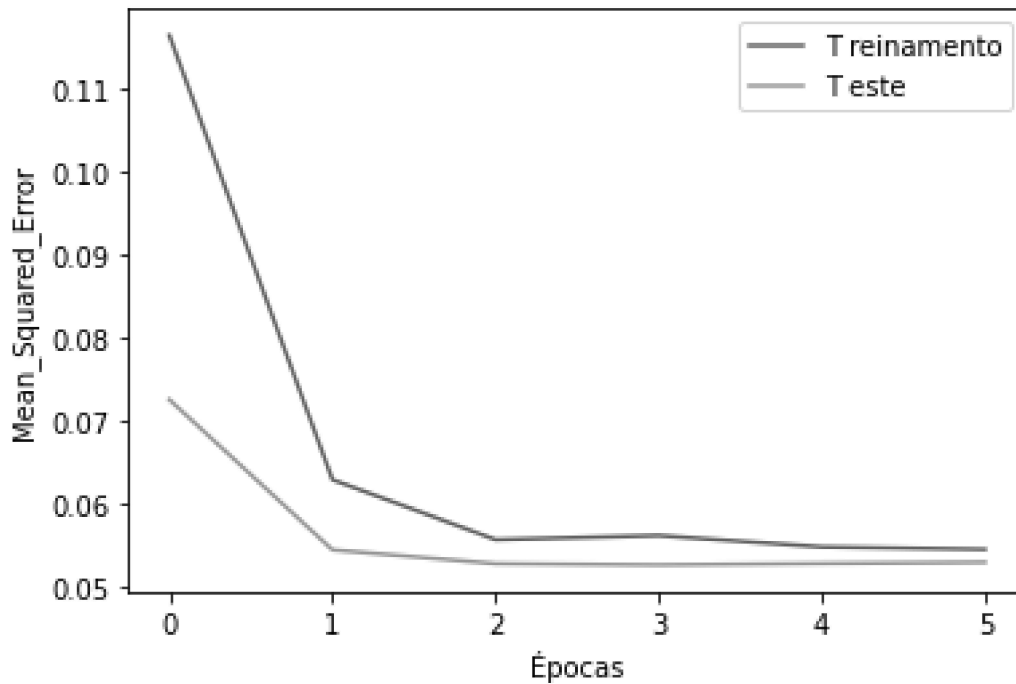


Figura 7.7 desempenho do modelo falta fase-terra 7º trecho

7.1.1.1 Resultados com variação das distâncias falta fase - terra

Outra informação muito relevante a ser extraída dos gráficos, é o fato de que para os trechos menores como os casos do trecho 1 e do trecho 5, bem como os trechos próximos das fontes, seus erros tendem a ser maiores em relação aos trechos com comprimentos maiores e distantes das fontes, parte disso é explicado pelo reduzido número de dados contidos nos trechos menores e outra explicação é a proximidade com que os trechos têm com as suas respectivas fontes, porque, quando a falta ocorre próxima à fonte, o impacto na localização da falta pode ser maior, pois a corrente elétrica flui para a fonte, e esse fenômeno pode interferir nos sinais elétricos medidos pelos sensores. O que faz com que resulte em maiores distorções nos dados de entrada dos modelos de redes neurais, afetando assim, a precisão da localização da falta. Como visto nos gráficos anteriores e nas tabelas 7.2, 7.3.

Tabela 7.2 – Localização de faltas variando as distâncias fase-terra

Trecho	C (km)	Dist. Reais (km)	Resposta da RNA treinada Rf = 250Ω	Erro absoluto (km)	Erro relativo (%)
1	30	15	14,80	0,20	1,33
2	40	50	49,91	0,085	0,17
3	30	85	84,95	0,05	0,058
4	70	35	34,80	0,20	0,57
5	30	15	14,81	0,19	1,26
6	40	50	49,92	0,08	0,16
7	40	90	89,95	0,05	0,055

Tabela 7.3 – Erros do modelo da rede neural artificial

Rf	Erro mínimo	Erro máximo	Erro médio	Desvio padrão
250Ω	0,050	0,20	0,122	0,071

A figura 7.8 apresenta o histograma de distribuição dos erros absolutos em km das distâncias para faltas fase-terra a 50% de cada trecho previstos pelo modelo de rede neural artificial, e ilustra de forma clara a variação dos erros absoluto entre os trechos.

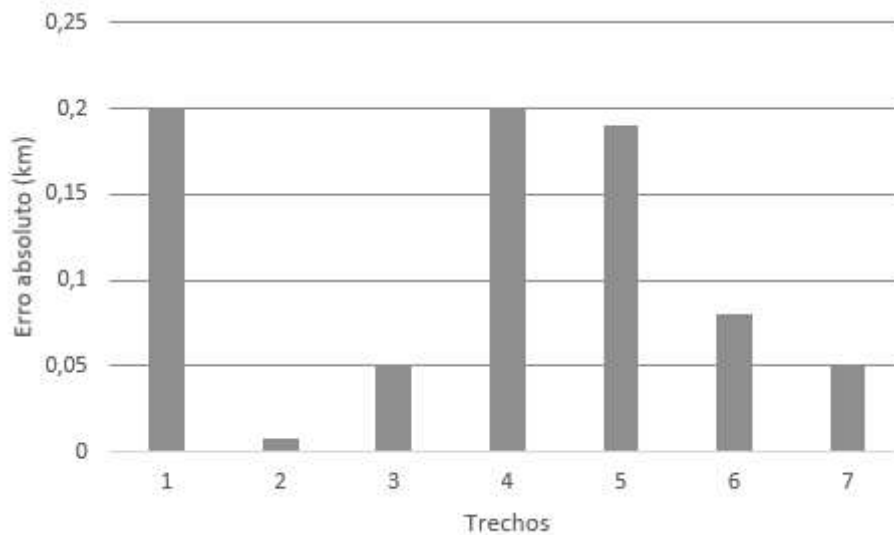


Figura 7.8 – histograma dos erros absolutos das distâncias para faltas fase-terra a 50% de cada trecho

7.1.2 Faltas fase – fase - fase

Resistência de defeito a 250 ohm

Distância = 50% de cada trecho

Durante uma falta trifásica, ocorre um evento mais grave do que um curto-circuito monofásico. O curto-circuito trifásico resultar em correntes mais elevadas e maiores esforços mecânicos nas estruturas da linha de transmissão.

De maneira similar ao descrito no subcapítulo 7.1 para a seleção dos neurônios que compõem as camadas intermediárias, foram testados vários modelos até que fosse encontrado aquele que proporcionasse os melhores resultados, como mostrado na Tabela 7.4.

Tabela 7.4 – Teste de diferentes arquiteturas de redes neurais artificiais

Trecho	m	n	Épocas	Convergência
1	10	10	15	Não
	13	13	15	Não
	20	20	5	Sim
2	10	10	15	Não
	16	13	5	Sim
	20	20	5	Não
3	16	16	15	Não
	18	18	15	Não
	20	20	5	Sim
4	16	16	15	Não
	18	18	15	Não
	20	20	5	Sim
5	16	16	15	Não
	18	18	15	Não
	20	20	5	Sim
6	10	10	15	Não
	13	13	5	Sim
	25	25	15	Não
7	16	16	15	Não
	18	18	15	Não
	30	30	5	Sim

m = primeira camada intermediária

n = segunda camada intermediária

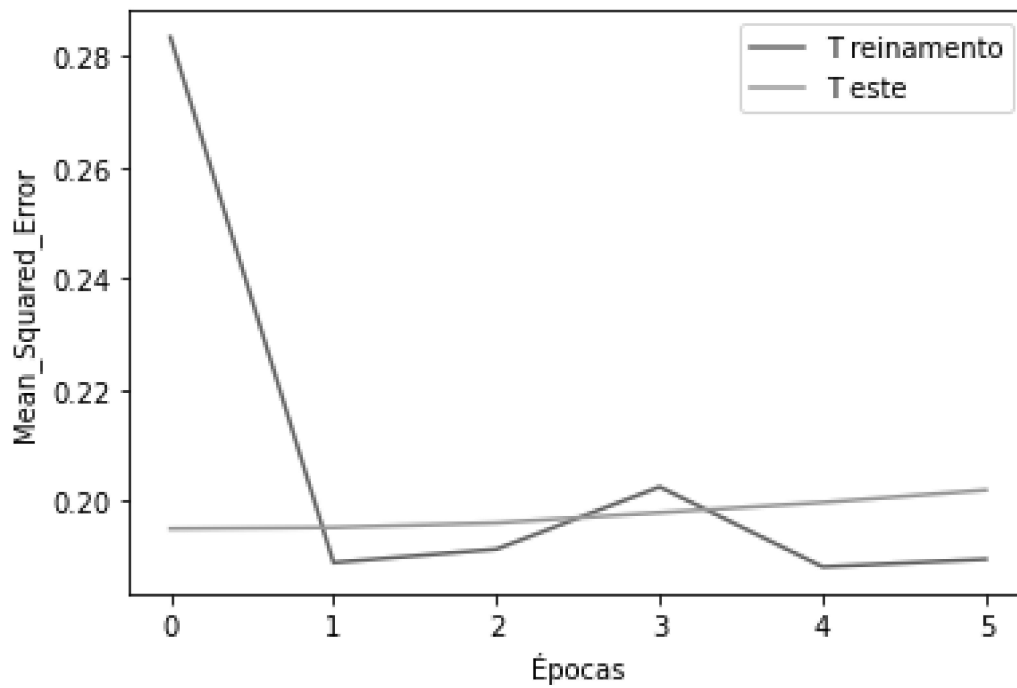


Figura 7.9 desempenho do modelo falta 3fases-terra 1º trecho

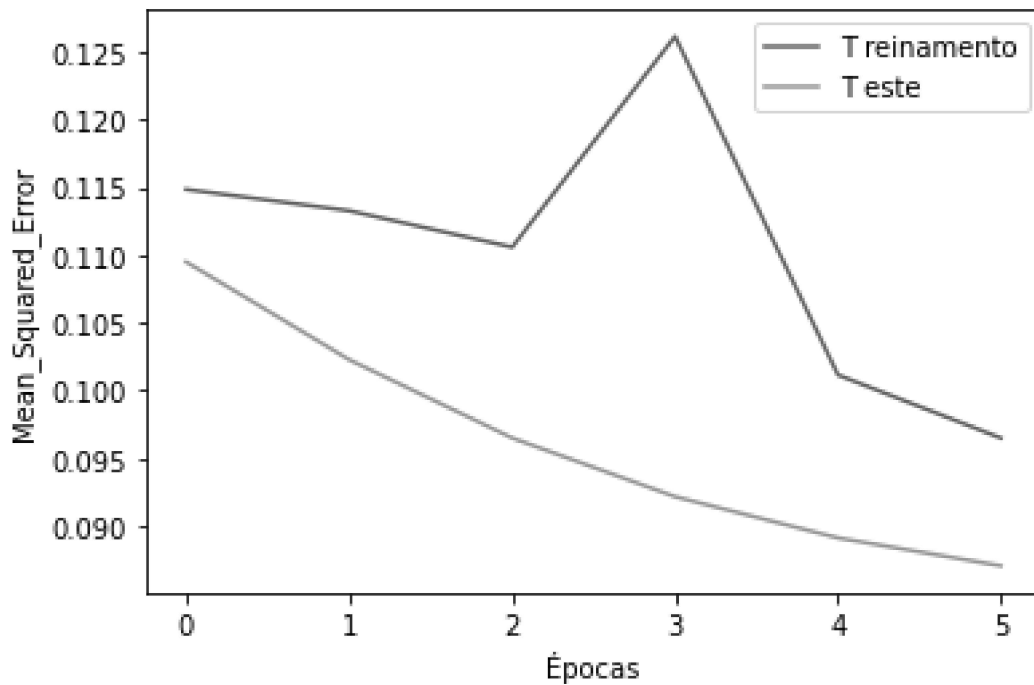


Figura 7.10 desempenho do modelo falta 3fases-terra 2º trecho

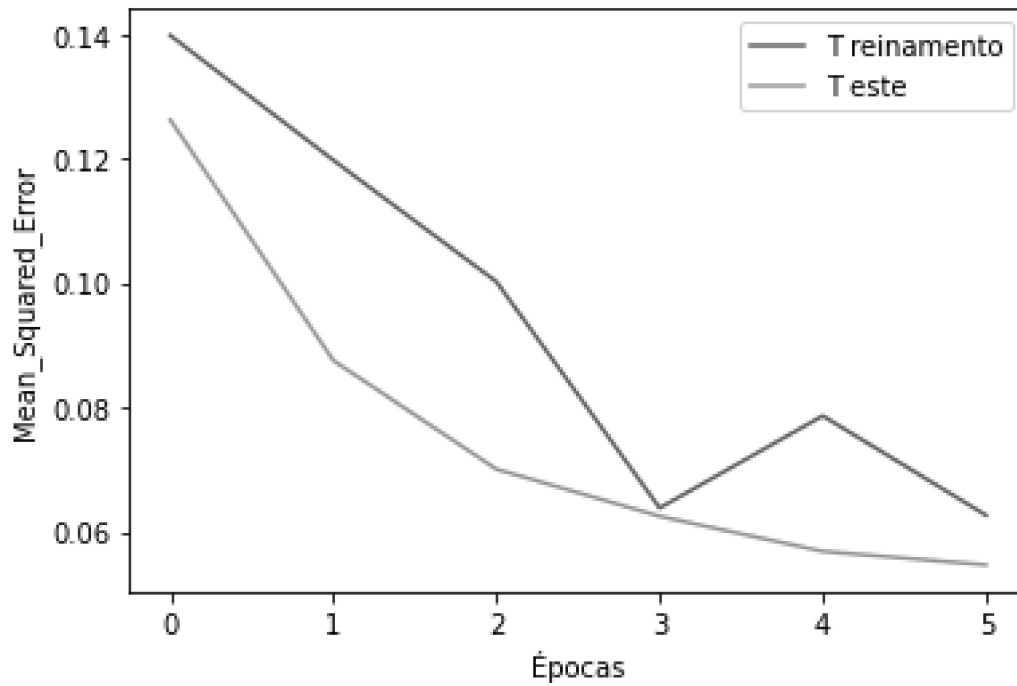


Figura 7.11 desempenho do modelo falta 3fases-terra 3º trecho

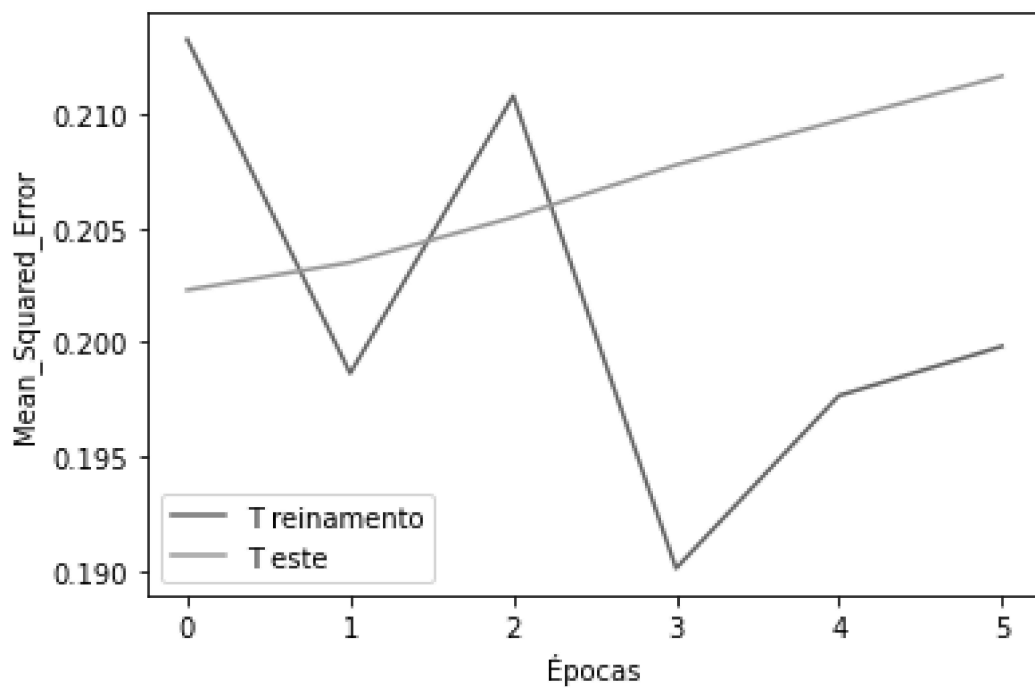


Figura 7.12 desempenho do modelo falta 3fases-terra 4º trecho

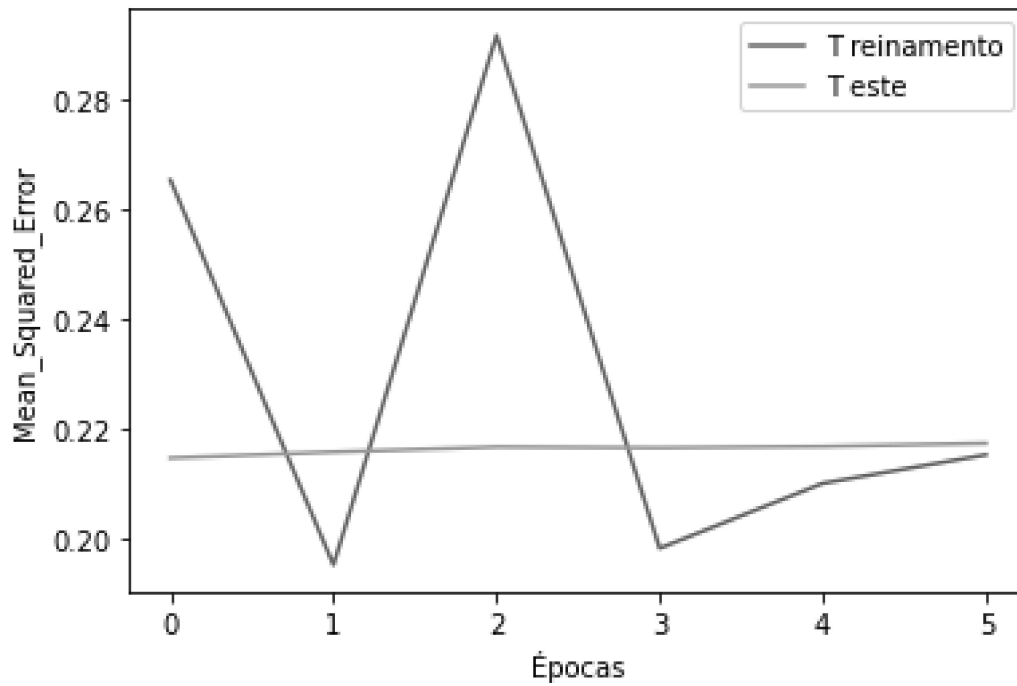


Figura 7.13 desempenho do modelo falta 3fases-terra 5º trecho

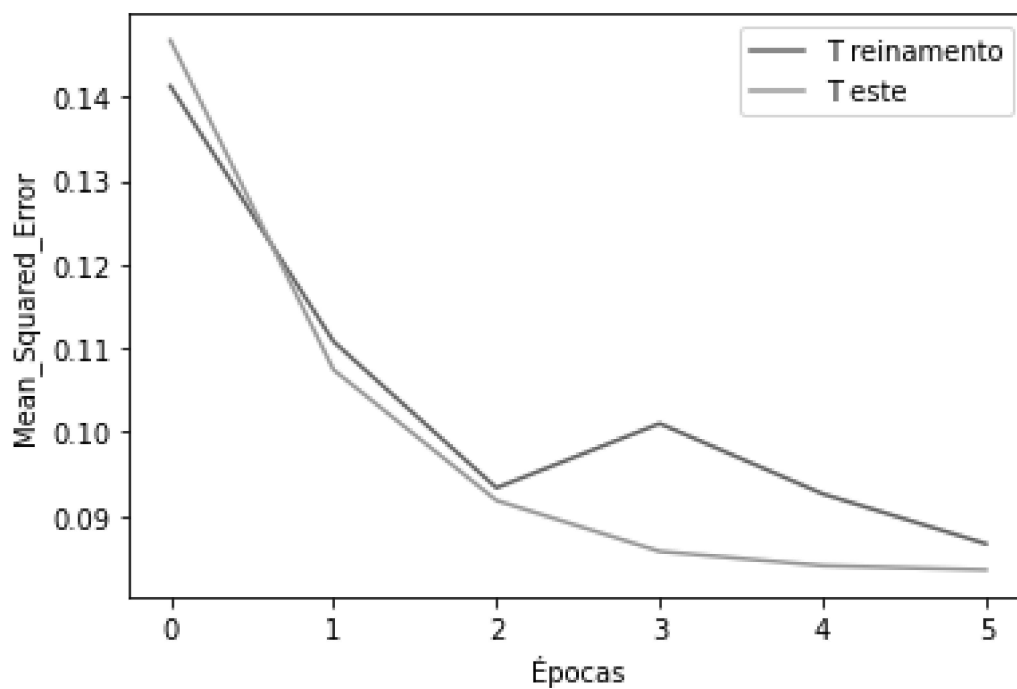


Figura 7.14 desempenho do modelo falta 3fases-terra 6º trecho

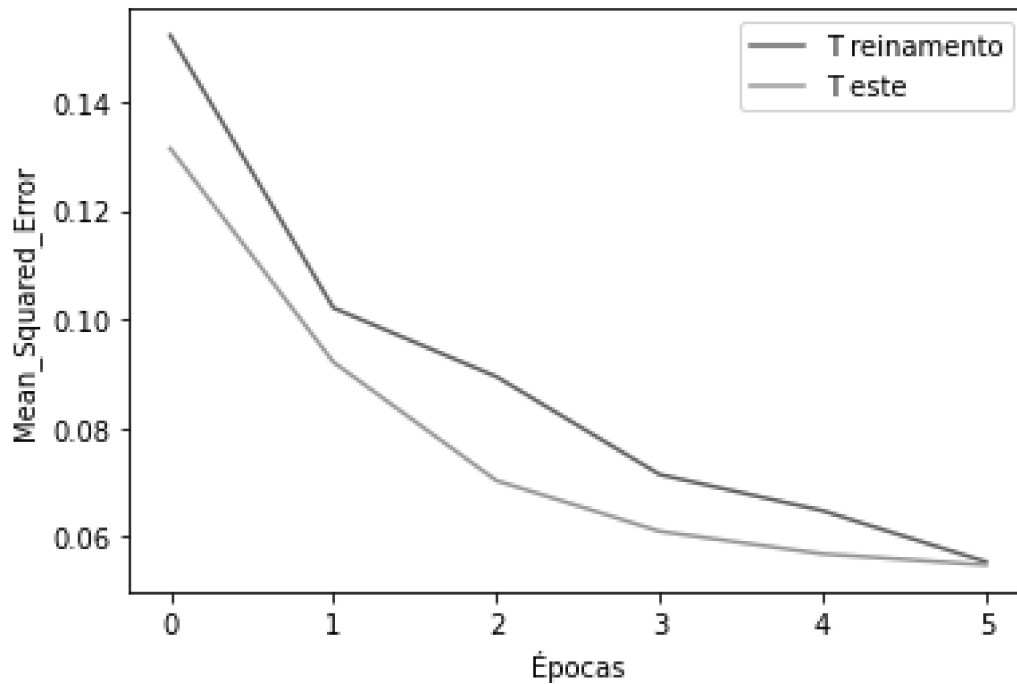


Figura 7.15 desempenho do modelo falta 3fases-terra 7º trecho

Os trechos representados nas figuras 7.8, 7.11 e 7.12 apresentaram valores de erros relativamente superiores aos dos outros trechos, pelos mesmos motivos mencionados na seção 7.1.1.1. Trechos de menor comprimento tendem a produzir menores quantidades de dados, enquanto o oposto ocorre com os trechos mais longos. Para corroborar os gráficos, os valores dos erros também estão apresentados numericamente nas Tabelas 7.5 e 7.6.

7.1.2.2 Resultados com variação das distâncias falta fase – fase - fase – terra.

Tabela 7.5 – Localização de faltas variandos as distâncias 3fases-terra

T	C (km)	Dist. Reais (km)	Resposta da RNA treinada Rf = 250Ω	Erro absoluto	Erro relativo (%)
1	30	15	14,79	0,201	1,34
2	40	50	49,91	0,087	0,174
3	30	85	84,94	0,06	0,07
4	70	35	34,79	0,21	0,6
5	30	15	14,78	0,22	1,46
6	40	50	49,90	0,10	0,2
7	40	90	89,95	0,05	0,055

Tabela 7.6 – Erros do modelo da rede neural artificial

Rf	Erro mínimo	Erro máximo	Erro médio	Desvio padrão
250Ω	0,050	0,22	0,132	0,074

A figura 7.16 apresenta o histograma de distribuição dos erros absolutos em km das distâncias para faltas fase-fase-fase a 50% de cada trecho previstos pelo modelo de rede neural artificial.

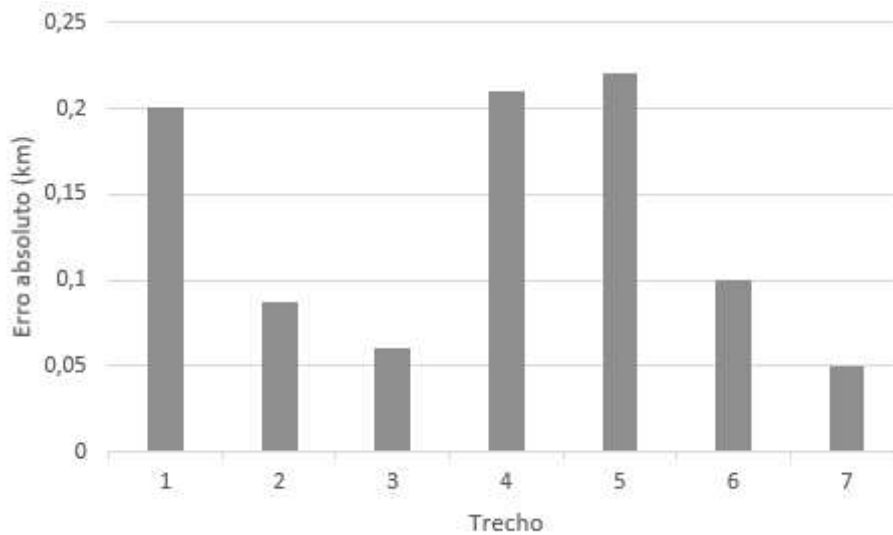


Figura 7.16 – histograma dos erros absolutos das distâncias para faltas fase-fase-fase a 50% de cada trecho

Após os treinamentos, os testes apresentaram respostas satisfatórias diante dos problemas apresentados, e o erro médio quadrático (*mean squared error*) foi menor que 10% para todos os parâmetros, incluindo tipo de faltas, resistência e distância dos trechos, conforme mostrado nas Tabelas 7.1 e 7.5. Apesar de ter sido utilizada uma quantidade ligeiramente menor de dados em alguns trechos, eles foram extremamente eficientes e precisos para o processo de localização de ocorrências de defeitos nos sistemas das linhas de transmissão de energia elétrica. Os modelos mostraram-se eficazes.

8 CONCLUSÃO E PORPOSTAS PARA TRABALHOS FUTUROS

Na presente dissertação foi desenvolvido uma técnica de localização de faltas em sistemas de linhas de transmissão de energia elétrica.

Para implementar a técnica, uma linha de transmissão foi simulada no software ATP e um algoritmo foi desenvolvido no MATLAB com a finalidade de automatizar o processo de simulações, gerando dados faltosos e não faltosos do sistema. Além disso, outro algoritmo também foi desenvolvido no MATLAB para gerar um banco de dados e armazenar os dados provenientes das simulações.

Através da revisão bibliográfica e da análise dos resultados de estudos apresentados, foi possível constatar que a utilização das redes neurais oferece uma série de benefícios, sendo assim, é forte candidata para auxiliar os modelos mais tradicionais relativamente aos métodos de localização de faltas em linhas de transmissão. A técnica é capaz de lidar com diferentes tipos de faltas, apresenta boa precisão e rapidez na identificação de problemas, e pode ser facilmente implementada em sistemas de monitoramento em tempo real.

Apesar da complexidade da técnica, os modelos propostos mostraram - se eficazes e promissores, pois, os resultados refletem o sucesso obtido. Os modelos obtiveram menos de 10% de erros ao longo do processo de todos os testes. Além da localização também foram abordados os temas da detecção de faltas e a classificação do tipo de faltas, nesses dois ultimos os modelos também apresentaram um ótimo desempenho como mostrado nas tabelas 6.1 e 6.2.

No entanto, alguns desafios ainda precisam ser superados para que a técnica seja amplamente utilizada. Dentre eles, destacam-se a necessidade de uma grande quantidade de dados para treinamento de uma rede neural, e a complexidade na configuração dos parâmetros e hiperparametros dos modelos de redes neurais.

Os modelos apresentaram maiores dificuldades na localização de faltas em trechos mais curtos, devido ao número reduzido de amostras disponíveis nesses trechos. Como resultado, os erros também foram maiores nessas áreas.

Vale ressaltar que, embora as redes neurais não sejam os melhores modelos para dados estruturados ou modulares, os modelos propostos ainda assim obtiveram resultados encorajadores, com menos de 10% de erros nos testes.

No presente trabalho, utilizou-se a técnica para predição. Para trabalhos futuros, espera-se que novos estudos aprofundem a pesquisa sobre a aplicação da técnica para a previsão de curtos circuitos em linhas de transmissão.

REFERÊNCIA

Anees A. S, M. Jamil and M. Rizwan, "**Fault location scheme on a three phase transmission line using synchronized phasor measurement**," 2012 IEEE 5th India International Conference

Bookman, 2001. D. Spoor and Jian Guo Zhu, "**Improved single-ended traveling-wave fault-location algorithm based on experience with conventional substation transducers**," in IEEE Transactions on Power Delivery, vol. 21, no. 3, pp. 1714-1720, July 2006, doi: 10.1109/TPWRD.2006.878091.

C. K. Y. Zerahny, L. K. Yun, W. J. K. Raymond and K. T. Mei, "**Fault Classification and Location in Three-Phase Transmission Lines Using Wavelet-based Machine Learning**," 2020 8th International Conference on Intelligent and Advanced Systems (ICIAS), Kuching, Malaysia, 2021, pp. 1-5, doi: 10.1109/ICIAS49414.2021.9642641.

Da Silva I. N, Spotti D. H, Flauzino R. A, **REDES NEURAIS ARTIFICIAIS**, Artliber, Sao Paulo 2010

D. C. Huynh, T. H. Truong and M. W. Dunnigan, "**Fault Location on Series-Compensated Transmission Lines of a Power System**," 2021 IEEE 2nd International Conference on Smart Technologies for Power, Energy and Control (STPEC), Bilaspur, Chhattisgarh, India, 2021, pp. 1-6, doi: 10.1109/STPEC52385.2021.9718737.

Di Santo S. G, **Localização de faltas em linhas de transmissão com múltiplas derivações utilizando dados de somente dois terminais**. Dissertação (mestrado em engenharia elétrica) Universidade de São Paulo, 2009.

Di Santo S. G. e Pereira C. E.M, "**Metodo de localizagao de falhas para linhas de transmissao nao transpostas com derivagoes**", PES T&D 2012, 2012, pp. 18, doi: 10.1109/TDC.2012.6281477.

Filipe, S.S Huilman, M. A. Maurfcio, N. A. P. Alfredo, J. S. C. Francisco, M. A. Fernando "**Localizagao de Faltas em Linhas de Transmissão pelo M etodo**

das Componentes Fundamentais" 2020 sociedade brasileira de automatica, DOI:10.48011/sbse.v1 i1.2418

Fan. R, T. Yin, R. Huang, J. Lian and S. Wang, "**Transmission Line Fault Location Using Deep Learning Techniques**," 2019 North American Power Symposium (NAPS), 2019, pp. 1-5, doi: 10.1109/NAPS46351.2019.9000224.

Haykin, S. **Redes Neurais: princípios e práticas**. Porto Alegre: Bookman, 2001.

Hinge T. P and S. S. Dambhare, "**Novel fault location algorithm for transmission line using synchronized measurements**," 2016 IEEE/PES Transmission and Distribution Conference and Exposition (T&D), 2016, pp. 1-6, doi: 10.1109/TDC.2016.7519874

Kazakidis S. A, A. I. Kokkosis, K. P. Moustiris and A. G. Paliatsos, "**Electricity consumption prognosis with the combination of smart metering and artificial neural networks**," 8th Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MEDPOWER 2012), 2012, pp. 1-6, doi: 10.1049/cp.2012.2013.

K. Jia, T. Bi, Z. Ren, D. W. P. Thomas and M. Sumner, "**High Frequency Impedance Based Fault Location in Distribution System With DGs**," in IEEE Transactions on Smart Grid, vol. 9, no. 2, pp. 807-816, March 2018, doi: 10.1109/TSG.2016.2566673.

Livani. H, C. Y. Evrenosoglu, "**A Machine Learning and Wavelet-Based Fault Location Method for Hybrid Transmission Lines**," in IEEE Transactions on Smart Grid, vol. 5, no. 1, pp. 51-59, Jan. 2014, doi: 10.1109/TSG.2013.2260421.

Louro, Antonio Henrique Figueira, Michelle Magalhaes Mendonga, and Adilson Gonzaga. "**CLASSIFICAÇÃO DE TOMATES UTILIZANDO REDES NEURAIS ARTIFICIAIS.**" Ciencia e Tecnologia De Alimentos: Pesquisa e Praticas Contemporaneas - Volume 2, n.d., 352-61. doi:10.37885/210805897.

Mamig M. S e M. Arkan, "**Algoritmo de localizagao de falhas baseado em FFT para linhas de transmissao**", 2011 7ª Conferencia Internacional de Engenharia Eletrica e Eletronica (ELECO), 2011, pp. I-71-I-75.

Mazon. A.J, I. Zamora, J.F. Minambres, M.A. Zorrozua, J.J. Barandiaran, K. Sagastabeitia, **"A new approach to fault location in two-terminal transmission lines using artificial neural networks,"** Electric Power Systems Research, V 56, Issue 3, 2000, P 261-266, ISSN 0378-7796.

Mohsen Tabari, Javad Sadeh, **"Fault location in series-compensated transmission lines using adaptive network-based fuzzy inference system",** Electric Power Systems Research, Volume 208, 2022.

Mousinho, Renata, and Cruz Sergio. **"Redes Neurais e Transtornos De Aprendizagem: Rastreo De Pessoas Com Dislexia."** Anais Do Simposio Brasileiro De Informatica Na Educacao, 2009.

Sayari Das, Shiv P. Singh, Bijaya K. Panigrahi, **Transmission line fault detection and location using Wide Area Measurements,** Electric Power Systems Research, V 151, 2017, P 96-105, ISSN 0378-7796.

Soufiane B, Nouredine B, Azzeddine , Bousaadia B, Karim A, **"Deep learning through LSTM classification and regression for transmission line fault detection, diagnosis and location in large-scale multi-machine power systems".** Measurement. V 177, 2021, 109330, ISSN 0263-2241.

Senger, E.C. & Jr, Giovanni & Filho, Francisco & Nakagomi, Renato & Wehrend, Harald & Zhang, Biao. (2021). **ALGORITMO PARA LOCALIZAQAO DE FALTAS EM LINHAS DE TRANSMISSAO.**

SOUZA, D. P. M, **Localizagao de Faltas em Linha de Transmissao de Energia pela Analise da Fungao de Autocorrelagao 2018.** Dissertação (mestrado em engenharia eletrica) Universidade Federal Fluminense, Volta Redonda, p 26, 2022.

Stevenson Jr., William D. **Elementos de análise de sistemas de potência.** São Paulo: McGraw-Hill, 1982.

S. Yan, J. Xu, X. Ju, Z. Wu, H. Fan and P. Cao, "**A Novel Fault Location Method for Transmission Lines Based on Progressive Kurtosis in ± 800 kV Kunbei Converter Station,**" 2022 5th International Conference on Energy, Electrical and Power Engineering (CEEPE), Chongqing, China, 2022, pp. 437-441, doi: 10.1109/CEEPE55110.2022.9783294.

Wang. C, B. Zhang, G. Li, L. Yan and J. Zhao, "**One practical method of fault location for transmission lines based on dual-voltage fault components,**" 2011. International Conference on Electrical and Control Engineering, 2011, pp. 30653068, doi: 10.1109/ICECENG.2011.6057787.