

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
FACULDADE DE ENGENHARIA ELÉTRICA E BIOMÉDICA
CURSO DE ENGENHARIA ELÉTRICA

Luiz Henrique Pinto Assunção

**Otimização de Dispositivos Baseados em
Cristais Fotônicos Usando Métodos em
Machine Learning**

BELÉM – PARÁ
2022

Luiz Henrique Pinto Assunção

Otimização de Dispositivos Baseados em Cristais Fotônicos Usando Métodos em Machine Learning

Trabalho de Conclusão de Curso submetido
ao curso de Engenharia Elétrica da Facul-
dade de Engenharia Elétrica e Biomédica do
Instituto de Tecnologia da Universidade Fe-
deral do Pará, como requisito parcial para a
obtenção do Grau de Bacharel em Engenha-
ria Elétrica.

Orientador: Prof. Dr. Victor Dmitriev
Coorientador: Prof. Dr. Ronaldo de Freitas
Zampolo

BELÉM – PARÁ
2022

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

A851o Assunção, Luiz Henrique Pinto.
Otimização de dispositivos baseados em cristais fotônicos
usando métodos em machine learning / Luiz Henrique Pinto
Assunção. — 2022.
93 f. : il. color.

Orientador(a): Prof. Dr. Victor Dmitriev
Coorientador(a): Prof. Dr. Ronaldo de Freitas Zampolo
Trabalho de Conclusão de Curso (Graduação) - Universidade
Federal do Pará, Instituto de Tecnologia, Faculdade de Engenharia
Elétrica, Belém, 2022.

1. Nanofotônica. 2. Inteligência Artificial. 3. Otimização.
4. Modelagem Inversa. I. Título.

CDD 620.5

Luiz Henrique Pinto Assunção

Otimização de Dispositivos Baseados em Cristais Fotônicos Usando Métodos em Machine Learning

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção de grau de Bacharel em Engenharia Elétrica, aprovado pela banca examinadora que atribuiu o conceito Excelente. Belém – Pará, 21 de janeiro de 2022.

Prof. Dr. Victor Dmitriev
Orientador

Prof. Dr. Ronaldo de Freitas Zampolo
Coorientador

Prof. Dr. Rodrigo M. e S. de Oliveira
Membro da Banca Examinadora

Prof. Dr. Miércio C. de A. Neto
Membro da Banca Examinadora

Prof. Dr. Miércio C. de A. Neto
Diretor da Faculdade de Engenharia
Elétrica e Biomédica

BELÉM – PARÁ
2022

DEDICATÓRIA

*Dedico este trabalho a todos que se dedicam à ciência para mudar
o Mundo em que vivemos.*

AGRADECIMENTOS

Este trabalho foi possível em decorrência de muitas vivências que obtive durante a minha vida.

Agradeço ao meu Orientador Victor Dmitriev e Coorientador Ronaldo Zampolo pela orientação neste trabalho.

Agradeço aos amigos do Laboratório de Nanofotônica e Nanoeletrônica pelo suporte no desenvolvimento da pesquisa que resultou no presente trabalho e do contato com o Programa de Pós-Graduação em Engenharia Elétrica. Em especial: Gianni Portela, Geraldo Melo e Wagner Castro.

Agradeço a todos os meus professores que me proporcionaram um grande aprendizado ao decorrer da graduação. Em especial: Roberto Menezes, Washington Sousa, Claudiomiro Barbosa e Thiago Mota.

Agradeço a minha família: Deuza (minha mãe), José João (meu pai), Natália e Jéssica (minhas irmãs).

Agradeço aos meus amigos que me acompanharam na graduação, em especial: Rodrigo Dutra, Edilberto Oliveira, Marcio Muniz, Abel Massunanga.

Agradeço aos meus amigos do projeto de divulgação científica Clube de Astronomia do Pará (CAP), pelos esforços e companheirismo na divulgação científica e combate às *pseudociências*.

Agradeço, também, a mim mesmo pelo esforço e intensa curiosidade em procurar saber *como as coisas funcionam*, característica que se manifestou em mim desde que me entendo por gente, o que me permitiu a sempre investigar novos desafios.

Esse trabalho teve o suporte do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), da Pró-Reitoria de Pesquisa (PROPESP) e da Universidade Federal do Pará (UFPA).



EPÍGRAFE

*“O nitrogênio em nosso DNA, o cálcio em nossos dentes,
o ferro em nosso sangue, o carbono em nossas tortas de maçã...
Foram feitos no interior de estrelas em colapso,
agora mortas há muito tempo.
Nós somos poeira das estrelas.”
(Carl Sagan – COSMOS, 1980)*

RESUMO

Neste trabalho, é estudado um método de otimização e modelagem inversa de dispositivo nanofotônico para aplicação em sistemas de telecomunicações. Nas últimas décadas, o avanço tecnológico possibilitou um melhor entendimento da interação da luz com a matéria em escala nanométrica. Nesse contexto, surge a nanofotônica, uma área que vem atraindo muitas pesquisas, sobretudo, na fabricação de novos dispositivos que operam na faixa do terahertz. Se por um lado as simulações eletromagnéticas possibilitam a construção de novos dispositivos, por outro isso se torna um processo demorado à medida que aumenta a complexidade da resposta eletromagnética dessas nanoestruturas. Uma nova abordagem que vem surgindo é a modelagem inversa, isto é, modelar a geometria do dispositivo a partir da premissa de uma resposta em frequência com parâmetros ótimos de operação do dispositivo. Neste trabalho, é demonstrado o poderoso uso das redes neurais profundas no processo de modelagem inversa de dispositivos nanofotônicos. No total, quatro dispositivos foram submetidos ao método. Os resultados foram satisfatórios, em especial, quando a abordagem foi aplicada para dois circuladores de ressonância dipolo e quadrupolo baseados em cristal fotônico. Em ambos, a rede neural profunda teve êxito em realizar a modelagem inversa dos dispositivos, trabalho concluído em menos de um mês após a implementação. O método foi aplicado também a dois dispositivos baseados em grafeno, entretanto, o procedimento não obteve um resultado satisfatório. Neste último caso, a otimização ainda pode ser melhorada para os referidos dispositivos, por exemplo, investigando mais variáveis (da geometria ou do material) que podem não estar sendo consideradas no banco de dados da rede neural profunda. É importante como o aprendizado profundo está revolucionando muitas áreas da tecnologia e, no contexto da nanofotônica, também tem se mostrado uma ferramenta poderosa para o *design* de nanoestruturas.

Palavras-chave: Nanofotônica. Inteligência Artificial. Otimização. Modelagem Inversa.

ABSTRACT

In this work, was developed optimization and inverse modeling method of nanophotonics devices for telecommunication system applications. In recent decades, technological advances have enabled a better understanding of the light-matter interaction at the nanometer scale. In this context, nanophotonics emerges as an area that has attracted a lot of research, especially in the design of new devices that operate in the terahertz range. As the electromagnetic simulations allow the study and design of these devices, on the other hand, that is a time-consuming process as their complexity increase. A new approach known as inverse modeling has been emerging in the last years, which consists in molding the device geometry from the optimal operating conditions in its frequency response. In this work, the powerful use of deep neural networks in the inverse modeling process of nanophotonic devices is demonstrated. Four devices were submitted by the method, and the results were particularly satisfactory when subjected to two dipole and quadrupole resonance circulators based on the photonic crystal. In both, the deep neural network was successful to predict the geometry of devices based solely on their target frequency response. The method to two graphene-based devices was also applied, and the results have shown that the procedure has not yet reached a satisfactory level. In the latter case, optimization can still be improved for those devices, for example, by investigating more variables (of geometry or material) that maybe are not being considered in the deep neural network database. It's significant how deep learning is revolutionizing many areas of technology and, in the context of nanophotonics, it has also proven to be a powerful tool for the design of nanostructures.

Key words: Nanophotonics. Artificial Intelligence. Optimization. Inverse Modeling.

LISTA DE ILUSTRAÇÕES

Figura 1 – Região de interesse em terahertz.	30
Figura 2 – Representação artística dos materiais. a) Cristal fotônico e seus arranjos dimensionais em 1D, 2D e 3D. b) Grafeno e sua estrutura hexagonal de um átomo de espessura.	31
Figura 3 – Ilustração de um dispositivo genérico divisor de potência. a) Divisor por 2. b) Divisor por 2 com uma porta isolada. a) Divisor por 3.	32
Figura 4 – Ilustração de um dispositivo genérico circulador de três portas. a) Incidência na porta 1. b) Incidência na porta 2. c) Incidência na porta 3.	33
Figura 5 – Estudo da resposta em frequência para um dispositivo de três portas. a) Para a <i>porta 1</i> . b) Para a <i>porta 2</i> . c) Para a <i>porta 3</i>	34
Figura 6 – Diagrama de Venn simplificado dos níveis hierárquicos da inteligência artificial.	35
Figura 7 – Neurônio biológico.	36
Figura 8 – Desenho de laminação cortical de Santiago Ramon y Cajal mostrando uma seção transversal da rede neural biológica (córtex humano) com os neurônios dispostos em múltiplas camadas.	37
Figura 9 – Diagrama de blocos de um neurônio artificial.	38
Figura 10 – Funções de ativação. a) Linear. b) Linear. c) Linear. d) Linear. e) ReLu. f) Leaky ReLu.	39
Figura 11 – Redes <i>Multilayer Perceptron (MLP)</i> . a) Rede MLP padrão. b) Rede neural profunda.	41
Figura 12 – Principais classes de redes neurais. a) Rede feedforward. b) Rede recorrente.	41
Figura 13 – Aprendizado Supervisionado.	42
Figura 14 – Descida do gradiente.	43
Figura 15 – Divisão das amostras do banco de dados.	44
Figura 16 – Modelagem convencional no COMSOL Multiphysics ®.	47
Figura 17 – Esquemático dos tipos de modelagens.	48
Figura 18 – Procedimento de contrução do banco de dados.	51
Figura 19 – Esquema básico da alimentação do banco de dados na rede neural.	53
Figura 20 – Arquitetura da rede neural profunda detalhada.	53
Figura 21 – Algoritmo de otimização.	55
Figura 22 – Diagrama de blocos simplificado do algoritmo de otimização.	56
Figura 23 – Fatores de qualidade avaliados na resposta em frequência.	57
Figura 24 – Funcionamento dos circuladores. a) Modo dipolo. b) Modo quadrupolo	58

Figura 25 – Geometria do cristal fotônico.	59
Figura 26 – Resposta em frequência desejada.	61
Figura 27 – Resposta em frequência do circulador dipolo após a modelagem inversa.	63
Figura 28 – Resposta em frequência do circulador quadrupolo após a modelagem inversa.	64
Figura 29 – Geometria dos circuladores após a otimização. a) Geometria base. b) Geometria final do circulador dipolo. c) Geometria final do circulador quadrupolo.	64
Figura 30 – Evolução da função custo para o circulador dipolo.	65
Figura 31 – Evolução da função custo para o circulador quadrupolo.	66
Figura 32 – Divisores de potência suas respectivas distribuições do campo eletromagnético para excitação na porta 1. a) Divisor $\mathcal{T}\sigma_1$. b) Divisor $\mathcal{T}\sigma_2$.	81
Figura 33 – Variáveis geométricas do divisor vertical $\mathcal{T}\sigma_1$.	82
Figura 34 – Variáveis geométricas do divisor horizontal $\mathcal{T}\sigma_2$.	82
Figura 35 – Resposta em frequência desejada.	84
Figura 36 – Resultado da resposta em frequência do divisor vertical $\mathcal{T}\sigma_1$.	85
Figura 37 – Resultado da resposta em frequência do divisor horizontal $\mathcal{T}\sigma_2$.	86
Figura 38 – Parametrização da geometria base do cristal fotônico.	88

LISTA DE TABELAS

Tabela 1 – Performance de cada arquitetura de rede em relação ao banco de dados inicial i_0 de cada circulador.	60
Tabela 2 – Comparação do desempenho de otimização dos divisores.	65
Tabela 3 – Fatores de qualidade avaliados para o circulador de ressonância dipolo.	67
Tabela 4 – Fatores de qualidade avaliados para o circulador de ressonância quadrupolo.	68
Tabela 5 – Performance de cada arquitetura de rede em relação ao banco de dados inicial i_0 de cada divisor.	84
Tabela 6 – Comparação dos parâmetros do divisor vertical.	86
Tabela 7 – Comparação do desempenho de otimização dos divisores.	87
Tabela 8 – Parâmetros otimizados do circulador dipolo.	89
Tabela 9 – Parâmetros otimizados do circulador quadrupolo.	89
Tabela 10 – Hiperparâmetros da rede neural.	90
Tabela 11 – Divisão do banco de dados.	90
Tabela 12 – Configurações do computador.	90

LISTA DE ABREVIATURAS E SIGLAS

Adam	Adaptive Moment Estimation
Adadelta	Adaptive Delta
Adagrad	Adaptive Gradient
ANN	Artificial Neural Network
API	Application Programming Interface
CI	Circuito Integrado
CNN	Convolutional Neural Networks
DNN	Deep Neural network
Eq.	Equação
Fig.	Figura
GAN	Generative Adversarial Network
GHz	Gigahertz
Loop	Conjunto de simulações
MEF	Método dos Elementos Finitos
MLP	Multilayer Perceptron
MSE	Mean Squared Error
Nadam	Nesterov-accelerated Adaptive Moment Estimation
PBG	Photonic Band Gap
ReLU	Rectified Linear Unit
RNA	Redes Neurais Artificiais
RMSprop	Root Mean Square Propagation
SGD	Stochastic Gradient Descent
THz	Terahertz

LISTA DE SÍMBOLOS

i	Instância
i_0	Banco de dados inicial
$[\mathbf{Y}]$	Tensor de parâmetros geométricos
$[\mathbf{X}]$	Tensor de resposta em frequência
$[\mathbf{Z}]$	Tensor de espectro desejado
$N_{[Y]}$	Normalização do tensor $[Y]$
$N_{[X]}$	Normalização do tensor $[X]$
$N_{[Z]}$	Normalização do tensor $[Z]$
$D_{[Y]}$	Desnormalização do tensor $[Y]$
F_c	Frequência central
ΔF_{ij}	Distância do ponto de inflexão da curva à F_c
$\Delta T1_{ij}$	Distância do ponto de inflexão da curva à -1 dB
$\Delta T2_{ij}$	Distância do ponto de inflexão da curva à -20 dB
f_1	Frequência inferior da curva mais interna a -15 dB
f_2	Frequência superior da curva mais interna a -15 dB
BW	Largura de banda
C	Função Custo
\mathcal{T}_{σ_1}	Divisor de potência com simetria vertical
\mathcal{T}_{σ_2}	Divisor de potência com simetria horizontal
dB	Decibel
$\varphi(\cdot)$	Função de ativação
w_{ij}	Peso sináptico atual
w_{ij}^+	Peso sináptico a ser atualizado

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Trabalhos Relacionados	26
1.2	Organização do Trabalho	28
2	REVISÃO BIBLIOGRÁFICA	29
2.1	Nanofotônica	29
2.1.1	Nanoestruturas	31
2.1.2	Divisores de Potência	32
2.1.3	Circuladores	33
2.1.4	Parâmetros-S e Resposta em Frequência	33
2.2	Aprendizado de Máquina	34
2.2.1	Redes Neurais Artificiais	35
2.2.2	Fundamentos Biológicos	36
2.2.3	Neurônio Artificial	37
2.2.4	Função de Ativação	38
2.2.5	Perceptron Multicamadas	40
2.2.6	Algoritmos e Processos de Aprendizagem	42
2.2.7	Algoritmo Backpropagation	44
3	MÉTODO PROPOSTO	47
3.1	Descrição do Problema	47
3.2	Otimização por Aprendizado Profundo	49
3.2.1	Construção do Banco de Dados	50
3.2.2	Rede Neural Profunda	53
3.2.3	Treinamento e Predição	54
3.2.4	Procedimento de Otimização	55
3.3	Fatores de Qualidade	57
3.4	Aplicação em Circulador	58
3.4.1	Arquitetura de Rede	59
3.4.2	Operação Ideal	60
4	RESULTADOS	63
4.1	Circulador	63
4.1.1	Fatores de Qualidade	67

5	CONSIDERAÇÕES FINAIS	69
5.1	Discussão	69
5.2	Conclusão	70
5.3	Sugestões Para Trabalhos Futuros	71
5.4	Trabalhos Desenvolvidos	71
	 REFERÊNCIAS	 73
6	APÊNDICES	81
6.1	APÊNDICE A – APLICAÇÃO EM DIVISOR DE POTÊNCIA	81
6.1.1	Arquitetura de Rede	83
6.1.2	Operação Ideal	84
6.1.3	Resultados	84
6.2	APÊNDICE B – CRISTAL FOTÔNICO: PARÂMETROS GERAIS	88
6.2.1	Características da Geometria	88
6.2.2	Hiperparâmetros da Rede Neural	90
6.2.3	Configurações da Máquina	90

1 INTRODUÇÃO

Muitas das descobertas na tecnologia foram possíveis a partir de um profundo entendimento das propriedades dos materiais [1]. Nesse sentido, os engenheiros aprenderam a fazer mais do que apenas manipular os materiais na sua forma bruta. Por exemplo, nas últimas décadas, uma nova fronteira foi alcançada a partir de vários estudos sobre o controle das propriedades óticas dos materiais. E é nesse contexto que surge a *nanofotônica*, um ramo da engenharia ótica e engenharia elétrica que estuda o comportamento e interação da luz nos materiais em escala nanométrica. Compreender esses fenômenos óticos possibilita, sobretudo, a fabricação de novos dispositivos para atuarem como Circuitos Integrados (CI's) em sistemas de telecomunicações.

Nos últimos anos, houve um significativo crescimento de estudos de novos dispositivos que operam na faixa do *Terahertz* (THz) do espectro eletromagnético (faixa de frequência de 0,1THz a 10THz), em especial, os dispositivos não-recíprocos tais como chaves, filtros, circuladores, divisores de potência e antenas [2–4]. A modelagem desses componentes não-recíprocos é feita através de simulações eletromagnéticas por meio do *Método dos Elementos Finitos* (MEF) [5], ao passo que quanto mais complexos forem os componentes, maior é o tempo e o consumo de recursos computacionais [6, 7].

Por outro lado, a Inteligência Artificial (IA) tem revolucionado várias áreas da tecnologia, onde muitas aplicações emergiram nos últimos anos, como visão computacional [8], carros autônomos [9], reconhecimento de fala [10], processamento de linguagem natural [11], reconhecimento facial [12], etc. Nesse sentido, a aplicação de Redes Neurais Artificiais (RNA) para a modelagem de dispositivos fotônicos cresceu significativamente nos últimos anos, tendo em vista que se aplicam muito bem aos problemas multivariáveis e não-lineares [6, 13–15].

Usar a inteligência artificial para modelar um dispositivo a partir de uma condição ideal (condições desejáveis de operação do dispositivo) é uma abordagem conhecida como *modelagem inversa* (ou ainda, *design inverso*) [14–16]. Neste caso, a modelagem direta é quando se executa o projeto de forma convencional, isto é, quando se constrói a geometria do dispositivo nanofotônico para então se obter a sua resposta em frequência (onde serão avaliados os parâmetros de operação e desempenho do dispositivo). Esse processo deve ser repetido tantas vezes quanto forem necessárias até chegar-se numa resposta em frequência considerada ótima. Na abordagem *modelagem inversa*, a geometria do dispositivo não é definida por primeiro. Ao invés disso, constrói-se uma resposta em frequência com condições ideais de operação do dispositivo. Desta forma, é montado um banco de dados com vários exemplos aleatórios de parâmetros de geometria associados com a respectiva

resposta em frequência. Após aprender, a partir do banco de dados, os princípios entre os parâmetros geométricos e a resposta em frequência, a rede neural poderá obter a geometria apropriada que está relacionada com a resposta em frequência desejada.

O presente trabalho se propôs à aplicação de Redes Neurais Profundas (do inglês: *Deep Neural Networks (DNNs)*) para a modelagem inversa de dispositivos nanofotônicos. Nessa abordagem, é mostrado como as DNNs podem agilizar o processo de *design* e fornecer uma capacidade de caracterização robusta e eficiente de nanoestruturas complexas com base em uma resposta em frequência desejada, em um tempo significativamente menor que os métodos convencionais. Os dispositivos submetidos a esse processo são dois circuladores baseados em cristais fotônicos (discutidos em [17]) e dois divisores de potência baseados em grafeno (discutidos em [18] (ver Apêndice 6)). Todos os dispositivos são não-recíprocos e operam na região do terahertz e subterahertz.

1.1 Trabalhos Relacionados

Muitos trabalhos no âmbito da modelagem inversa de geometria de dispositivos fotônicos foram desenvolvidos nos últimos anos. A proposta, em comum nesses estudos, é muito parecida com a que é abordada neste presente trabalho: utilizar redes neurais profundas para modelar a geometria de dispositivos fotônicos e nanofotônicos a partir de uma resposta em frequência com características desejadas de operação.

No estudo discutido em [14], os autores desenvolveram uma arquitetura de rede neural para a modelagem inversa de dispositivos nanofotônicos, ao mesmo tempo, em que visa solucionar o problema de não-unicidade da resposta eletromagnética. Isto é, no problema abordado pelos autores, a resposta eletromagnética do dispositivo estudado não é única, pois várias configurações de geometria de dispositivo podem resultar na mesma resposta eletromagnética. Essa categoria de problema torna muito difícil o treinamento de redes neurais a partir de um banco de dados muito extenso, pois gera conflitos nas instâncias de treinamento, como o utilizado pelos autores, contendo 500 mil instâncias. Como solução, os autores propuseram uma arquitetura de rede com duas estruturas características: *rede direta* e a *rede inversa*, ambas com uma única camada intermediária. A *rede direta* é responsável por mapear as relações da geometria com o espectro. Posteriormente, a rede *rede inversa* é alimentada com o espectro (concatenada com a saída da *rede direta*) e produz em sua saída as geometrias associadas. Desta forma, é demonstrado que a arquitetura de rede proposta tolera instâncias de treinamento não exclusivas explícitas e implícitas, além de fornecer uma maneira de treinar grandes redes neurais para o projeto de modelagem inversa de estruturas fotônicas complexas.

Em [13], os autores propuseram a resolução do *design inverso* de nanoestruturas em metasuperfícies definidas por alguns parâmetros através de uma rede neural bidirecio-

nal. Nesse sentido, foram desenvolvidas duas redes neurais profundas, sendo uma rede de previsão de geometria (chamada de *GPN*) e uma rede de previsão de espectro (chamada de *SPN*). A rede GPN prevê os parâmetros geométricos dado uma resposta espectral, enquanto a SPN mapeia esse espectro para com a geometria de entrada da estrutura estudada. Durante o processo de treinamento, a saída da GPN é alimentada na SPN (processo parecido com o estudo em [14]). Dado um par de treinamento composto por parâmetros geométricos e seu espectro correspondente, o objetivo é minimizar a perda entre o par de treinamento e as saídas da GPN e SPN. Após o treinamento da rede bidirecional, novos projetos de modelagem inversa com várias respostas espetrais desejadas podem ser gerados rapidamente alimentando os objetivos na GPN. Outro mecanismo que os autores implementaram foi uma rede neural que compõe uma *estrutura de pré-processamento*. Na avaliação dos autores, a inclusão do pré-processamento demonstrou um melhor desempenho quando avaliado com diferentes arquiteturas de rede.

Em [19], os autores implementaram Redes Neurais profundas (do inglês: *Deep Neural Network* (DNN)) para a modelagem inversa de um dispositivo divisor de potência por dois (1×2) baseado em cristal fotônico. Todo o processo foi implementado com o uso do *framework* TensorFlow, no ambiente da linguagem de programação Python. No referido estudo, os autores apresentam um exemplo de nanocavidade em uma heteroestrutura de cristal fotônico 2D. O objetivo da otimização é identificar as posições dessas cavidades de modo a maximizar os fatores de qualidade dado uma certa estrutura inicial. O banco de dados usado para alimentar a DNN foi na ordem de 20.000 instâncias e o *espaço de design* na ordem de 2^{400} . Nesse sentido, os autores demonstram que as DNNs, contando com sua capacidade de processar dados volumosos e de grande dimensão do *espaço de design* (conceito explicado à diante), tornaram-se uma arquitetura indispensável para o projeto de dispositivos fotônicos com alta complexidade geométrica.

Outra abordagem de modelagem inversa é discutida em [20], onde os autores introduzem os rápidos avanços nas técnicas de *machine learning* e suas aplicações no processo de modelagem e otimização das estruturas fotônicas. Assim, são avaliados diferentes *graus de liberdade* (do inglês: *Degree of Freedom* (DOF)), fator atribuído conforme a *liberdade* que as técnicas de modelagem permitem ao projetista. Todas as possibilidades de configurações e arranjos de geometria é definido como um *espaço de design* e o número de variáveis consideradas corresponde à dimensionalidade desse espaço.

Por exemplo, para modelagens com *soluções analíticas* ou *varredura paramétrica simples*, implica em um baixo DOF, pois as combinações dos parâmetros de *design* não permitem explorar um espaço grande de soluções. No sentido crescente do DOF, encontraram-se as soluções por meio do *design inverso* (escopo deste trabalho), onde o uso de técnicas em aprendizado de máquina permite explorar um espaço maior de soluções (isto é, crescimento da dimensionalidade do *espaço de design*). Nesse contexto,

são usados modelo discriminativo para capturar as relações entre parâmetros de *design* e respostas ópticas com quantidades reduzida de dados. Deve-se notar que, dado que múltiplas configurações de estruturas podem corresponder à mesma resposta ótica (problema da *não-unicidade*), de forma que um único modelo discriminativo não é capaz de mapear perfeitamente uma resposta ótica de volta a um conjunto único de parâmetros de *design*. São necessárias estratégias adicionais de treinamento se modelos discriminatórios forem usados para a otimização e *design*.

Quando o DOF continua crescendo, modelos generativos podem ajudar a reduzir a dimensionalidade do *espaço de design*, de forma a buscar relações entre parâmetros de *design* e respostas ópticas para maior otimização. Os modelos generativos podem ser aproveitados conjuntamente com modelos discriminativos, bem como algoritmos tradicionais de otimização para acelerar o processo de *design* ou localizar as soluções ideais globais. Uma abordagem desse alto DOF são as que utilizam o aprendizado não-supervisionado, fazendo o uso de Redes Neurais Adversárias Generativas (do inglês: *Generative Adversarial Network (GAN)*) [21].

1.2 Organização do Trabalho

Este trabalho está organizado como se segue.

O Capítulo 2 é dividido em duas principais partes: na Seção 2.1 é feita uma revisão dos conceitos básicos da nanofotônica; e na Seção 2.2 são apresentados os conceitos básicos de redes neurais artificiais.

No Capítulo 3 é vista a metodologia de aplicação do procedimento de modelagem inversa e otimização.

O Capítulo 4 mostra os resultados obtidos após a implementação do método de otimização proposto neste trabalho.

Por fim, no Capítulo 5, são mostradas as considerações finais desde trabalho, onde perpassa pela *discussão, conclusão e sugestão para trabalhos futuros*.

Em adicional, o Apêndice 6 é apresentado em duas partes: a Seção 6.1 mostra todo o estudo do presente trabalho aplicado a outros dois dispositivos divisores de potência baseados em grafeno; e a Seção 6.1 mostra o detalhamento dos parâmetros de geometria dos dispositivos baseados em cristal fotônico, os hiperparâmetros da Rede Neural e as configurações do computador no qual o ambiente de otimização foi executado.

2 REVISÃO BIBLIOGRÁFICA

Neste Capítulo, são apresentados os conceitos básicos da *nanofotônica* (Seção 2.1), no que tange o desenvolvimento de dispositivos fotônicos e nanofotônicos, e os conceitos básicos do *aprendizado de máquina* (Seção 2.2), onde são introduzidos os conceitos de inteligência artificial e redes neurais profundas.

2.1 Nanofotônica

A nanofotônica é uma área da engenharia elétrica e engenharia ótica que visa o estudo da compreensão da luz nos materiais em escala nanométrica, o que certamente apresenta desafios para a ciência, ao mesmo tempo em que possibilita inovações tecnológicas. Esse estudo parte de várias frentes, como a investigação de novas interações óticas, de novos materiais, de técnicas de fabricação, bem como a exploração de estruturas orgânicas e inorgânicas, ou ainda, estruturas quimicamente fabricadas, como cristais fotônicos e pontos quânticos e plasmônicos [22, 23].

Nesse contexto, a crescente experiência da fusão da *nanotecnologia* com a *fotônica*, que são as duas principais tecnologias do século XXI, tem desempenhado um papel fundamental em muitos sistemas emergentes: seja em telecomunicações, como comutação óptica, espectroscopia, laser e fibra ótica [24, 25]; seja em aplicações médicas, como uma nova forma de diagnóstico e tratamento de câncer e imagens médicas [26]; ou ainda, em energia renováveis, como em células fotovoltaicas de alta eficiência [27]. Este campo multidisciplinar também causou impacto na indústria, permitindo aos pesquisadores explorar novos horizontes em *design*, na engenharia, na química, na física dos materiais, dentre outros [1, 28].

O estudo das interações do fóton com a matéria em escalas incrivelmente pequenas, conhecidas como *nanoestruturas*, tem como finalidade, sobretudo, o desenvolvimento de dispositivos e componentes em escala nanométrica para diversas funcionalidades, tendo em vista a capacidade de realizar muitas funções novas com base na interação eletromagnética local. Na nanofotônica, os conceitos tradicionais de *interferência* e *difração* (mecânica clássica) não são mais aplicáveis, mas substituídos por alguns novos conceitos [1, 29]. É nesse contexto de pesquisas que tem surgido nas últimas décadas o interesse e desenvolvimento de novos dispositivos nanofotônicos para operar como circuitos integrados em sistemas de telecomunicações, especialmente, na região de interesse do *terahertz* [30, 31].

A região terahertz (THz) do espectro eletromagnético, também chamada de *THz GAP*, corresponde a uma faixa de comprimento de onda entre cerca de 3 mm e 30 μm (ou

ainda, de 100 GHz a 10 THz, em termos de frequência). Assim, como mostrado na Fig. 1, em frequências mais baixas encontra-se o *regime eletrônico* de radiação milimétrica ou de microondas, usado para aplicações em telecomunicações sem fio, como comunicação via satélite, rádio, etc. Para frequências mais altas, está o *regime fotônico* (ou ainda, *regime ótico*), onde dispositivos ópticos ativos, como lasers de semicondutores e diodos emissores de luz, geram luz visível e infravermelha para transmissão em fibra óptica e armazenamento de dados [30, 32].

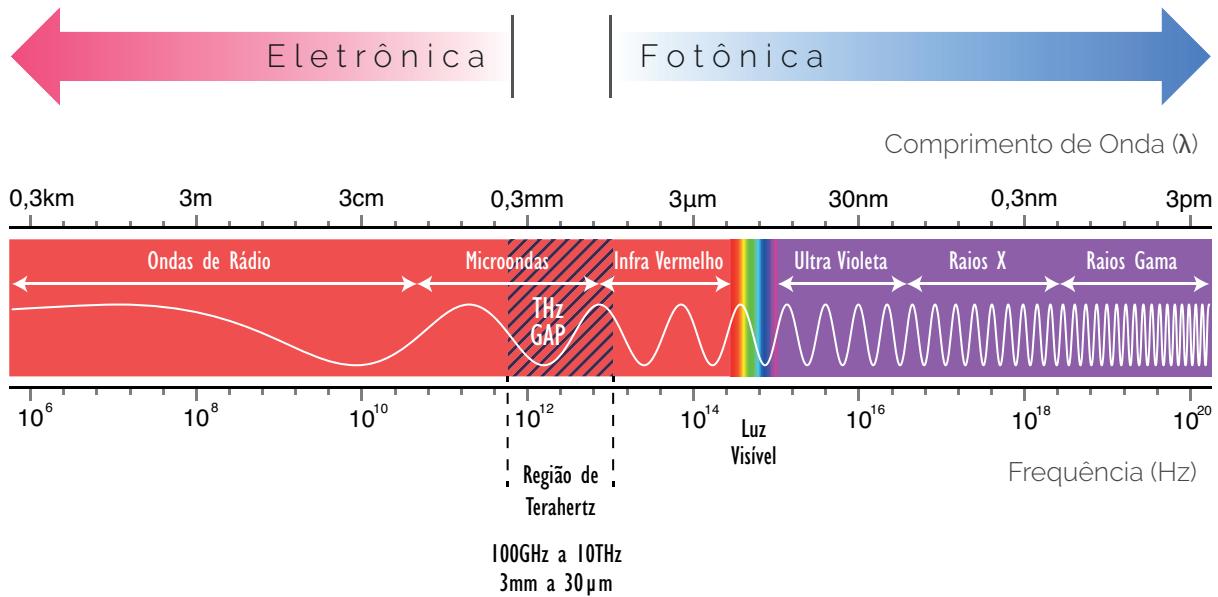


Figura 1 – Região de interesse em terahertz.

Fonte: do Autor.

Entre esses dois mundos está a lacuna da tecnologia THz, região pouco explorada tecnologicamente em comparação com a maior parte do restante do espectro eletromagnético, pois, embora seja simples gerar e manipular micro-ondas e radiação infravermelha, ainda existem desafios tecnológicos a serem superados para a geração e detecção da radiação THz [33]. Os *raios-T* (outro nome pelo qual a radiação THz é conhecida) têm aplicações potenciais em telecomunicações de última geração, incluindo as tecnologias de comunicação sem fio 6G e 7G [34], em sistemas de segurança, exames médicos e até mesmo na análise de obras de arte [32, 35].

Pelo fato da radiação THz ser do tipo *não ionizante*, isso a torna segura para o uso em humanos, posto que possa penetrar em muitos materiais visualmente opacos. Isso a torna potencialmente útil na varredura de segurança de armas escondidas ou explosivos. A radiação terahertz facilmente pode ser absorvida pela água [32], entretanto, pode penetrar por alguns milímetros o tecido biológico, o que confere a capacidade para os pesquisadores estudarem os fundamentos da biologia celular e molecular [36].

2.1.1 Nanoestruturas

Alguns materiais 2D apresentam um potencial tecnológico extraordinário para a engenharia de dispositivos e componentes nanoeletrônicos e nanofotônicos, de forma que interagem muito bem com a radiação eletromagnética na faixa do terahertz. Alguns exemplos desses materiais (nanoestruturas) são o *cristal fotônico* e o *grafeno* [37, 38], mostrados na Fig. 2.

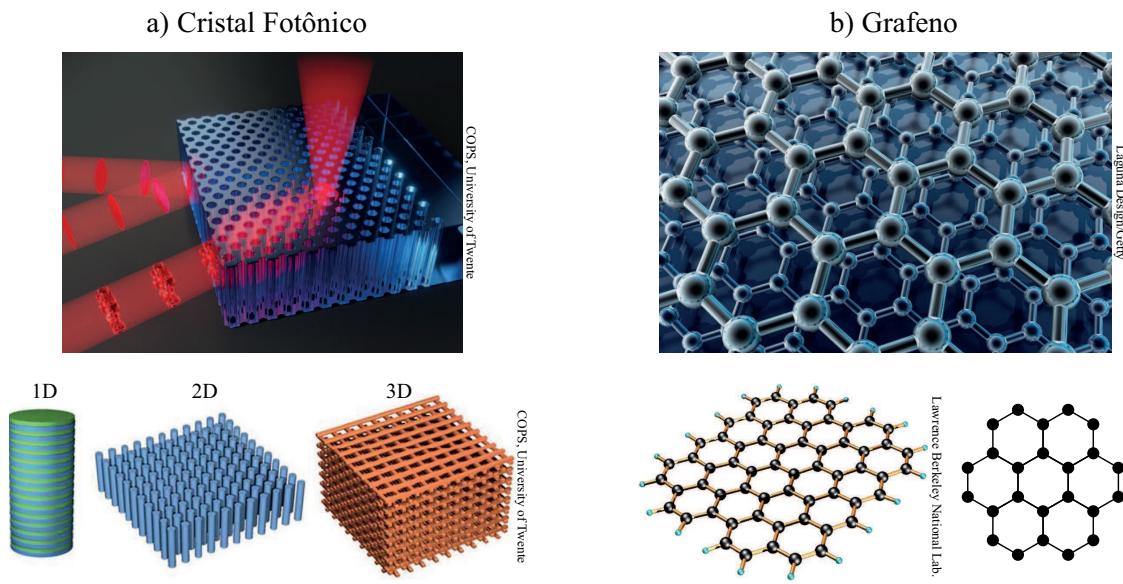


Figura 2 – Representação artística dos materiais. a) Cristal fotônico e seus arranjos dimensionais em 1D, 2D e 3D. b) Grafeno e sua estrutura hexagonal de um átomo de espessura.

Fonte: Isabelle Dumé (2021) [39]; Elizabeth Gibney (2018) [40] (Adaptado pelo Autor).

Os *cristais fotônicos* (do inglês: *Photonic Crystals (PhC)*) são nanoestruturas projetadas para afetar o movimento dos fótons, definindo bandas de energia permitidas e proibidas [22, 23]. Nesse sentido, os cristais fotônicos são compostos de nanoestruturas dielétricas e periódicas, com constante dielétrica alternada em uma, duas ou três dimensões para afetar a propagação de ondas eletromagnéticas dentro da estrutura. Como resultado dessa periodicidade, a transmissão da luz é absolutamente zero em certas faixas de frequência, o que é chamado de banda fotônica proibida (do inglês: *Photonic Band Gap (PBG)*) [41].

Ao introduzir os defeitos (pequenas modificações) nessas estruturas periódicas, a periodicidade é, portanto, a integridade da *banda fotônica proibida* é totalmente quebrada, o que permite controlar e manipular a luz [42] no material. Isso garante a localização da luz na região da banda fotônica proibida, o que possibilita ao desenvolvimento de dispositivos ópticos baseados em cristais fotônicos [43].

Um outro exemplo de nanoestrutura muito estudada é o *grafeno*, material 2D que mais vem sendo explorado nos últimos anos, sendo a substância mais fina já feita (contém apenas um átomo de espessura, como ilustrado na Fig. 2.b)), o mais forte e o de maior mobilidade, demonstrado até agora. O grafeno explora uma condutividade térmica recorde que, combinada com sua qualidade cristalina e eletrônica excepcionalmente alta, permitiu abrir um novo paradigma da física da matéria condensada. Essas combinações de propriedades são bastante peculiares do grafeno e não podem ser encontradas em nenhum outro material, ou ainda, sistema de materiais. Além disso, as propriedades óticas do grafeno, como o índice de refração, a absorção, velocidade de plasmon, dentre outras, podem ser ajustadas por meio de grade eletrostática ou pelo aumento do número de camadas. Isso implica que a permissividade do material pode ser alterada propositalmente, com a finalidade de que o grafeno possa se comportar como um material semicondutor, ou transparente, e até mesmo a propagação de plásmons em sua superfície [38].

Portanto, as nanoestruturas são diversas e suas características físicas são bastante estudadas para a fabricação de novos dispositivos, a exemplo de divisores de potência, chaves, circuladores, isoladores, dentre outros. Por conseguinte, metodologias computacionais têm sido amplamente empregadas para simular o comportamento funcional de estruturas nanofotônicas complexas que não possuem uma solução analítica.

2.1.2 Divisores de Potência

Um *divisor de potência* divide um sinal de entrada em dois ou mais sinais de saída. Como ilustrado na Fig. 3, um dado dispositivo divisor de potência recebe um guia de onda externo incidente em uma porta (*porta 1*) e o divide para as outras portas. Dependendo das características de projeto, pode ser interessante o isolamento alguma porta para a proteção de algumas partes do circuito (como ilustrado na Fig. 2.b)). Também é interessante, enquanto característica de projeto, o isolamento contra possíveis reflexões do sinal para a própria porta excitada (*porta 1*).

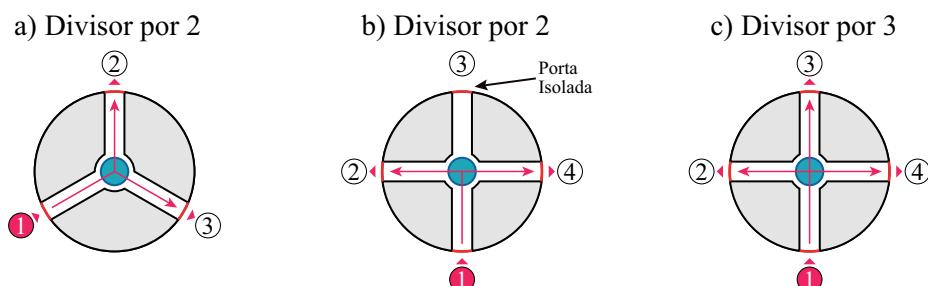


Figura 3 – Ilustração de um dispositivo genérico divisor de potência. a) Divisor por 2. b) Divisor por 2 com uma porta isolada. a) Divisor por 3.

Fonte: do Autor.

2.1.3 Circuladores

Um *circulador* é um dispositivo passivo, cuja principal característica é fazer o sinal de entrada sair para a próxima porta. Por exemplo, como mostrado na Fig. 4, o sinal injetado na *porta 1* interage com a cavidade ressonante central e segue para sair pela *porta 2*, mas o contrário não acontece, isto é, o sinal injetado na *porta 2* não retorna para a *porta 1*. Essas características dos circuladores os tornam muito úteis para aplicações, por exemplo, que envolvem o acoplamento de transmissores e receptores que compartilham uma antena comum, ao mesmo tempo em que isola o receptor do transmissor.

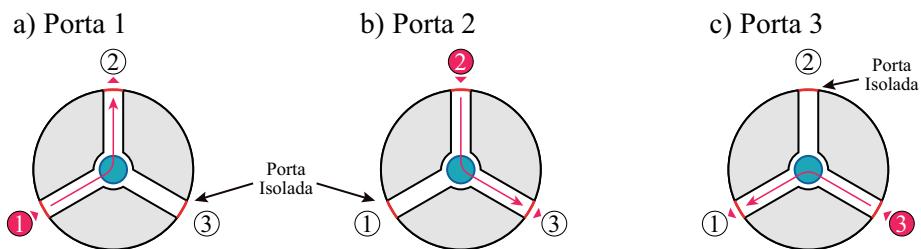


Figura 4 – Ilustração de um dispositivo genérico circulador de três portas. a) Incidência na porta 1. b) Incidência na porta 2. c) Incidência na porta 3.

Fonte: do Autor.

Assim, os circuladores são descritos como componentes *não recíprocos*, ou seja, o sinal na *porta 1* sai predominantemente da *porta 2* (Fig. 4.a)), o sinal na *porta 2* sai pela *porta 3* (Fig. 4.b)) e o sinal injetado pela *porta 3* sai para a *porta 1* (Fig. 4.c)). Em um dispositivo recíproco, a mesma fração do sinal que flui da *porta 1* para a *porta 2* ocorreria para o sinal fluindo na direção oposta, da *porta 2* para a *porta 1*.

2.1.4 Parâmetros-S e Resposta em Frequência

No estudo dos dispositivos nanofotônicos, uma maneira de avaliar o seu desempenho é através de sua resposta em frequência. Assim, pode-se mensurar os coeficientes de transmissão relacionados a cada porta. Por exemplo, para o circulador ilustrado na Fig. 4, a resposta em frequência está hipoteticamente ilustrada na Fig. 5. Para as portas 1, 2 e 3 há três curvas sendo avaliadas: curva de *transmissão* (transmissão do sinal para a próxima porta), curva de *isolamento* (isolamento do sinal para a porta que não seja a receptora) e curva de *reflexão* (isolamento de reflexão do sinal para a própria porta emissora).

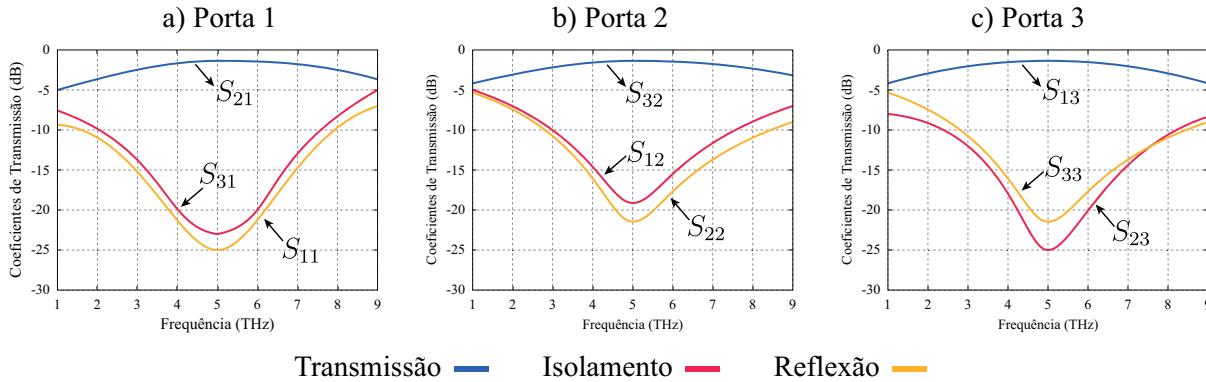


Figura 5 – Estudo da resposta em frequência para um dispositivo de três portas. a) Para a porta 1. b) Para a porta 2. c) Para a porta 3

Fonte: do Autor.

As aplicações em altas frequências tornaram-se cada vez mais predominantes com o avanço da eletrônica e da fotônica. Em circuitos de baixa frequência, por exemplo, parâmetros como tensão e corrente podem ser facilmente avaliados. Assim, os *parâmetros-Y* e os *parâmetros-Z* (admitância e impedância, respectivamente) podem ser usados para descrever uma rede. No entanto, para altas frequências, os *parâmetros-S* são melhores aplicáveis para estudar um sistema multiporta, de forma que eles definem as ondas refletidas em termos das ondas incidentes nesses sistemas [44]. Os parâmetros-S podem ser agrupados na forma matricial, conhecida como *matriz de espalhamento*, onde a ordem n da matriz S representa o número de portas e cada elemento S_{ij} , representa o coeficiente de transmissão ou reflexão do sinal estudado entre as portas. Por exemplo, o elemento S_{32} representa o coeficiente de transmissão avaliando um sinal da *porta 2* para a *porta 3*.

Para uma rede de três portas, como o dispositivo na representado na Fig. 4: os parâmetros S_{11} , S_{22} e S_{33} representam as perdas por reflexão das portas 1, 2 e 3, respectivamente; os parâmetros S_{21} , S_{32} e S_{13} representam os coeficientes de transmissão do sinal; e os parâmetros S_{31} , S_{12} e S_{23} representam os isolamentos.

2.2 Aprendizado de Máquina

O aprendizado profundo (do inglês: *Deep Learning (DL)*) é uma das áreas mais recentes em inteligência artificial, a qual envolve a aplicação de um subconjunto de ferramentas e técnicas de aprendizado de máquina (do inglês: *Machine Lerarning (ML)*), permitindo que modelos computacionais compostos por múltiplas camadas de processamento aprendam a representação do dado em múltiplos níveis de abstração. Em termos práticos, isso possibilita aos sistemas aprender com dados, identificar padrões e tomar decisões com base no que foi aprendido [45]. Esse processo possibilitou, sobretudo nos últimos anos, a revolução de muitas áreas da tecnologia, por exemplo, em carros autônomos [9], proces-

samento de linguagem natural [11], reconhecimento facial e visão computacional [8]. Mas os avanços não são sentidos apenas pela indústria da tecnologia. As empresas e organizações também têm investido no uso de técnicas em aprendizado de máquina, no âmbito da *ciência de dados*, para aumentar o lucro de seus negócios, ao mesmo tempo em que diminuem os custos de produção [46, 47].

A inteligência artificial, propriamente dita, é apenas o produto final de um processo bem mais estruturado. Para fim de conceituação, a Fig. 6 esclarece a organização hierárquica das várias vertentes da inteligência artificial.

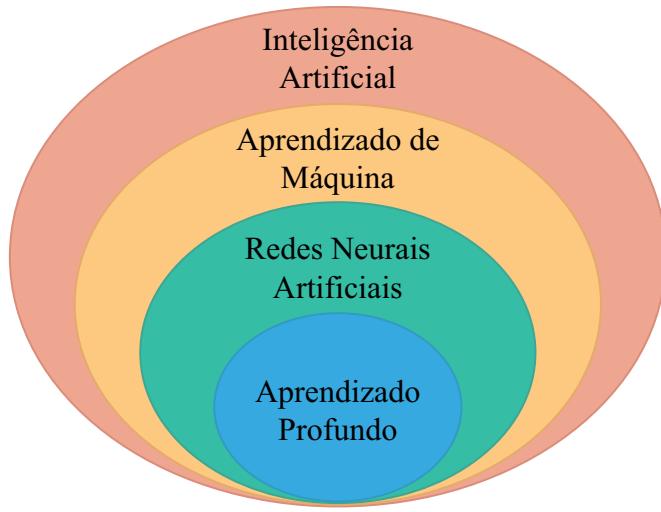


Figura 6 – Diagrama de Venn simplificado dos níveis hierárquicos da inteligência artificial.

Fonte: do Autor.

As Redes Neurais Artificiais (do inglês: *Artificial Neural Network (ANN)*) são o começo das pesquisas em inteligência artificial. Assim, elas são técnicas computacionais que partem de uma modelagem matemática inspiradas na estrutura neural biológica dos organismos inteligentes. Estruturas mais complexas das ANNs são as *Redes Neurais Profundas* (do inglês: *Deep Neural Networks (DNN)*). Elas são versáteis, poderosas e escaláveis, tornando-as ideais para lidar com tarefas de aprendizado de máquina grandes e altamente complexas, envolvendo problemas multivariáveis [48]. Os exemplos de aplicações estão muito presentes no dia-a-dia, como usados pelo *Google*: para a classificação de bilhões de imagens no mecanismo de pesquisa *Google Imagens*; para o melhoramento de tradução entre diversos idiomas do *Google Tradutor*; para o sistema de recomendação de vídeos do *Youtube* [49].

2.2.1 Redes Neurais Artificiais

As Redes Neurais Artificiais são uma das estruturas que compõe a base da inteligência artificial e tem raízes em disciplinas como neurociência, matemática, estatís-

tica, física, ciência da computação e engenharia. Suas aplicações podem ser encontradas em campos tão diversos quanto modelagem, análise de séries temporais, reconhecimento de padrões, processamento de sinais e controle. Elas são uma modelagem matemática-computacional dos neurônios biológicos humanos.

O estudo sobre RNAs existe há bastante tempo. Elas foram aboradas pela primeira vez em 1943 pelo neurofisiologista Warren McCulloch e pelo matemático Walter Pitts. Foi através do artigo *A Logical Calculus of the Ideas Immanent in Nervous Activity* [50] que McCulloch e Pitts apresentaram um modelo computacional simplificado de como os neurônios biológicos podem trabalhar juntos em cérebros de animais para realizar cálculos complexos usando lógica proposicional. Esta foi a primeira arquitetura de rede neural artificial.

2.2.2 Fundamentos Biológicos

Antes de discutirmos os neurônios artificiais propriamente ditos, é interessante pontuar algumas características de um neurônio biológico (representado na Fig. 7).

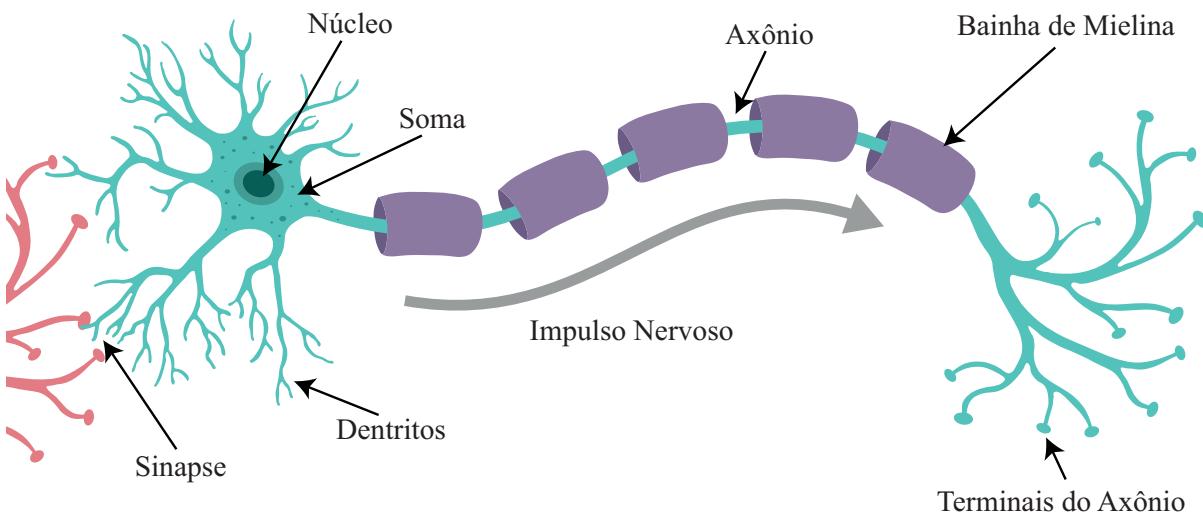


Figura 7 – Neurônio biológico.

Fonte: do Autor.

O neurônio é uma célula de aparência bastante característica encontrada, principalmente, em córtex cerebral de animais, composta por um corpo celular que contém o *núcleo* e a maioria dos componentes complexos da célula, e muitas extensões ramificadas chamadas *dendritos*, além de uma extensão longa chamada de *axônio*. O comprimento do axônio pode ser apenas algumas vezes maior do que o corpo celular ou até dezenas de milhares de vezes maior. Perto de sua extremidade, o axônio se divide em muitos ramos chamados *terminais*, e na ponta desses ramos estão estruturas minúsculas chamadas de *sinapses*, que estão conectadas aos dendritos de outros neurônios, e assim por diante,

formando uma *rede neural biológica*. Os neurônios biológicos recebem impulsos elétricos curtos (chamados de *sinais*) de outros neurônios por meio dessas sinapses. Quando um neurônio recebe um número suficiente de sinais de outros neurônios em alguns milissegundos, então, ele dispara seus próprios sinais [51, 52].

Individualmente, os neurônios biológicos parecem se comportar de uma forma bastante simples, entretanto, eles são organizados em uma estrutura em circuito, formando uma vasta rede de bilhões de neurônios, onde cada neurônio normalmente está conectado a outros milhares de neurônios. A arquitetura das redes neurais biológicas é constantemente objeto de pesquisas. Algumas partes do cérebro foram mapeadas por *Santiago Ramon y Cajal* (1899) e, como mostrado na Fig. 8, os neurônios costumam ser organizados em camadas consecutivas.

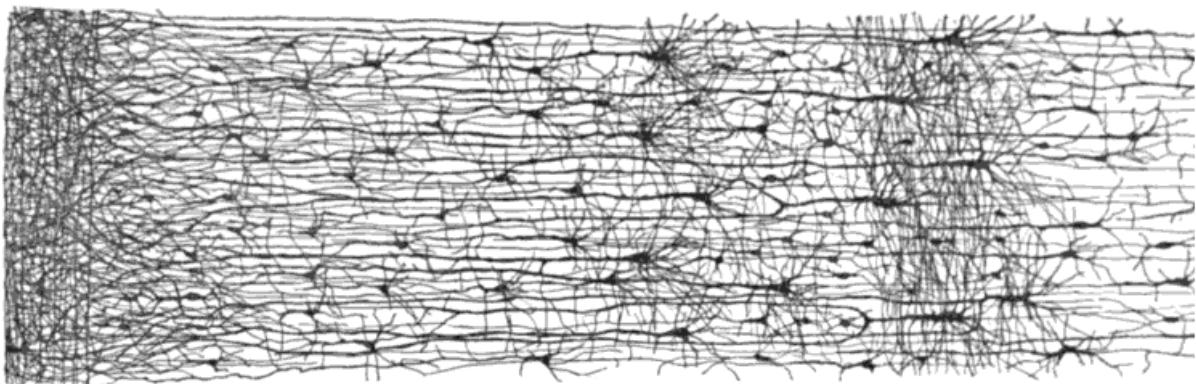


Figura 8 – Desenho de laminação cortical de Santiago Ramon y Cajal mostrando uma seção transversal da rede neural biológica (cortex humano) com os neurônios dispostos em múltiplas camadas.

Fonte: Santiago Ramon y Cajal (1899) [53].

2.2.3 Neurônio Artificial

Warren McCulloch e Walter Pitts propuseram um modelo muito simples do neurônio biológico, que mais tarde ficou conhecido como um *neurônio artificial* [50]: é constituído de uma ou mais entradas binárias (*nível lógico alto / nível lógico baixo*) e uma saída binária, de forma que neurônio artificial simplesmente ativa sua saída quando mais de um certo número de suas entradas estão ativas. Décadas de desenvolvimento e contribuição de vários pesquisadores, por fim, resultaram no modelo de neurônio artificial usado atualmente, conhecido como *perceptron* [47, 54].

Um neurônio artificial recebe em sua entrada sinais originários de outros neurônios. Essas conexões (*sinapses*) são ponderadas por elementos chamados de *pesos sinápticos*. Nessa linha, as entradas do neurônio podem ser *excitatórias ou inibitórias*. Assim, um neurônio artificial só poderá passar um sinal de saída para a próxima camada, caso suas

entradas somam um valor acima de um determinado valor limite, isto é, necessitam atingir um limiar de ativação (*função de ativação*) para o sinal ser propagado adiante. A Fig. 9 mostra um diagrama de blocos da representação matemática de um neurônio artificial perceptron.

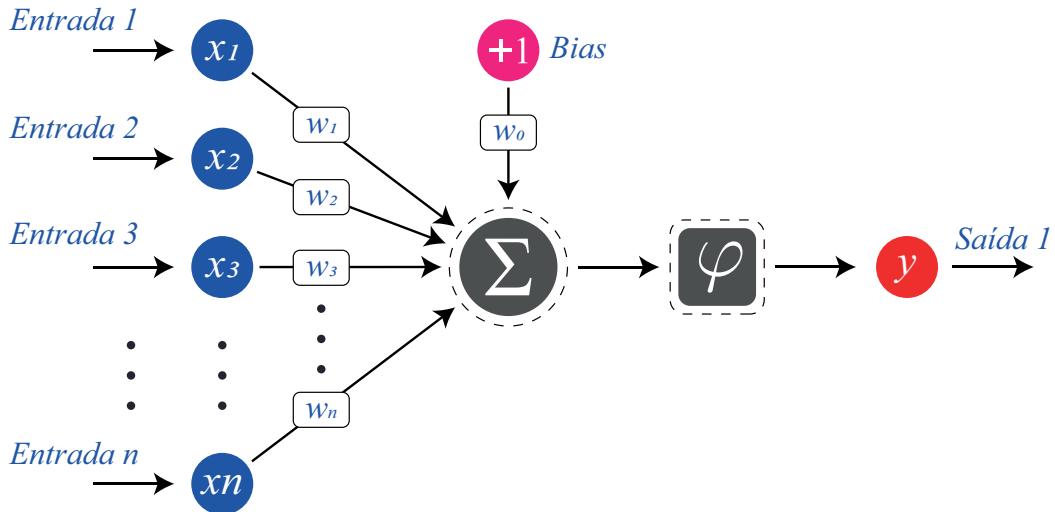


Figura 9 – Diagrama de blocos de um neurônio artificial.

Fonte: do Autor.

Na Fig. 9, x_1, x_2, \dots, x_n são as entradas; w_1, w_2, \dots, w_n são os pesos sinápticos; $\varphi(\cdot)$ é a função de ativação; e, por fim, y é o sinal de saída do neurônio. O somatório de todas as entradas do neurônio, ponderadas de pesos sinápticos, posteriormente submetidos a uma função de ativação, dão origem ao sinal de saída do neurônio. Nesse processo, o *Bias* (definido por b) é uma estrutura invariante de valor 1, ponderada de um peso w_0 . Em termos práticos, o *Bias* permite que o neurônio apresente uma saída não nula, ainda que todas as entradas sejam nulas [47, 54]. Matematicamente, um neurônio artificial j é modelado conforme as Eqs. 2.1 e 2.2.

$$v_j = \sum_{i=1}^n w_{ij} \cdot x_{ij} + b_j \quad (2.1)$$

Uma sinapse i causa um efeito pós-sináptico descrito por $w_i x_i$. A sinapse será *excitatória* se $w_i > 0$ e *inibitória* se $w_i < 0$. A Eq. 2.2 mostra a saída do neurônio.

$$y_j = \varphi(v_j) \quad (2.2)$$

2.2.4 Função de Ativação

A função de ativação $\varphi(\cdot)$ é uma transformação que será aplicada às entradas (ponderadas) antes do sinal ser enviado para a próxima camada de neurônios, sendo

essencial para determinar a saída desse neurônio, pois decidirá se o neurônio deve ser ativado ou não. A escolha da função de ativação tem um grande impacto na capacidade e no desempenho da rede neural, e diferentes funções de ativação podem ser usadas em diferentes partes do modelo. Existem muitos tipos de funções de ativação usadas em redes neurais, algumas das quais estão ilustradas na Fig. 10.

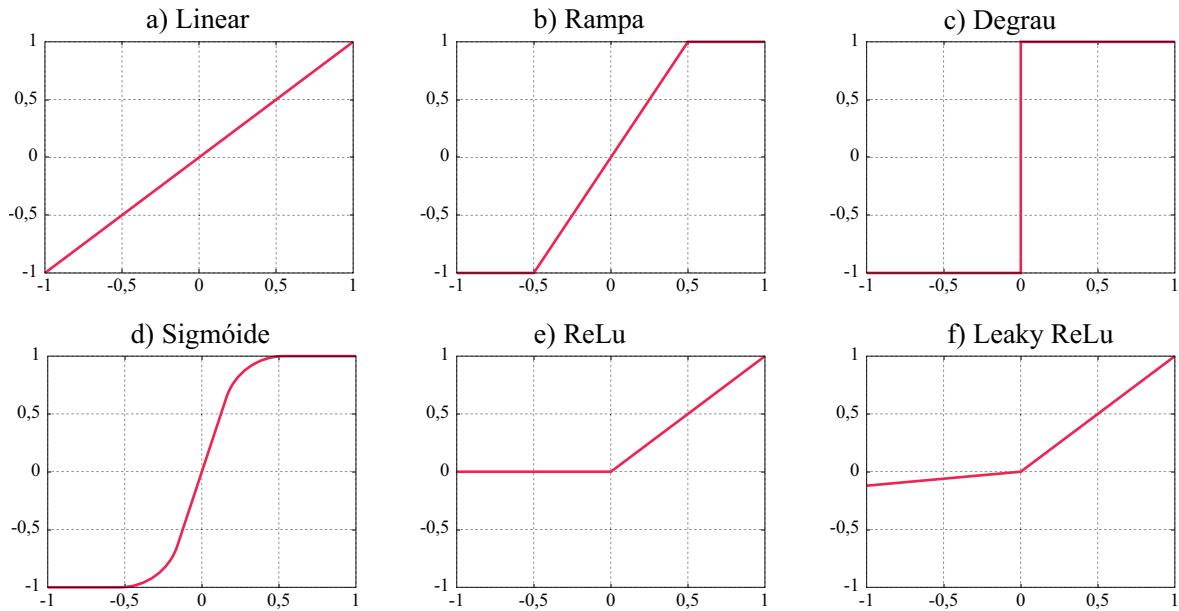


Figura 10 – Funções de ativação. a) Linear. b) Linear. c) Linear. d) Linear. e) ReLu. f) Leaky ReLu.

Fonte: do Autor.

A escolha da função de ativação é uma parte crítica do projeto da rede neural e a sua escolha é feita de acordo com a finalidade de previsão na saída do neurônio. Por exemplo, se a saída for um valor real, então o mais adequado é usar uma função de ativação *Linear* (Eq. 2.3), já que produz uma saída contínua e o algoritmo resultante torna-se o mesmo que uma regressão de mínimos quadrados [47]. Quando se deseja produzir uma saída contínua apenas em uma dada faixa de valores, o mais apropriado é a função *rampa* (Eq. 2.4) (ou ainda, *hard tanh*). A função *Degrau* (Eq. 2.5) é usada para representar uma saída de estados binários. A função *Sigmóide* (Eq. 2.6) é mais adequada para previsões que envolvam uma probabilidade de classes binárias.

Por fim, a função *ReLU* (do inglês: *Rectified Linear Unit*) é uma das funções de ativação mais utilizadas atualmente, principalmente em *redes neurais convolucionais*¹ ou *aprendizado profundo* [55, 56]. Assim, a função *ReLU* (Eq. 2.7) é retificada pela metade (na parte inferior), de forma que valores negativos tornam-se zero imediatamente, o que acaba gerando um problema por não mapear os valores negativos de forma adequada.

¹ Classe de rede neural artificial do tipo *feedforward*, muito utilizada em processamento e análise de imagens digitais.

Uma função que tenta resolver esse problema é a *Leaky ReLu* (Eq. 2.8), que é um tipo de função de ativação baseada em *ReLU*, mas tem uma pequena inclinação para valores negativos ao invés de retificá-los [47].

$$\varphi(x) = x \quad (2.3)$$

$$\varphi(x) = \begin{cases} +\lambda, & \text{se } x \geq +\lambda, \\ x, & \text{se } |x| < +\lambda, \\ -\lambda, & \text{se } |x| \leq -\lambda. \end{cases} \quad (2.4)$$

$$\varphi(x) = \begin{cases} +1, & \text{se } x \geq 0, \\ -1, & \text{se } x < 0. \end{cases} \quad (2.5)$$

$$\varphi(x) = \frac{1}{1 + e^{-x}} \quad (2.6)$$

$$\varphi(x) = \begin{cases} x, & \text{se } x \geq 0, \\ 0, & \text{se } x < 0. \end{cases} \quad (2.7)$$

$$\varphi(x) = \begin{cases} x, & \text{se } x \geq 0, \\ a \cdot x, & \text{se } x < 0. \end{cases} \quad (2.8)$$

Normalmente, todas as camadas ocultas usam a mesma função de ativação. A camada de saída normalmente usará uma função de ativação diferente das camadas ocultas e sua escolha depende muito do tipo de previsão exigida pelo modelo.

2.2.5 Perceptron Multicamadas

A arquitetura de uma rede neural é determinada pela forma com a qual os seus neurônios e camadas estão conectados [54]. O modelo *perceptron* de camada única (mostrado na Fig. 9) não pode produzir o tipo de desempenho que se espera de uma arquitetura de rede neural moderna, pois é limitado para a resolução de problemas linearmente separáveis, não sendo capaz de aproximar as relações complexas de entrada e saída que ocorrem em cenários de processamento de sinal da vida real [47, 54].

Enquanto que em uma rede de única camada, as entradas são diretamente mapeadas para a saída através de uma transformação da função de ativação, nas redes neurais com múltiplas camadas, as camadas de *entrada* e *saída* são separadas por um grupo de camadas *intermediárias*. Esse modelo descrito é uma classe de *rede feedforward* conhecida como *Multilayer Perceptron (MLP)* [47, 54].

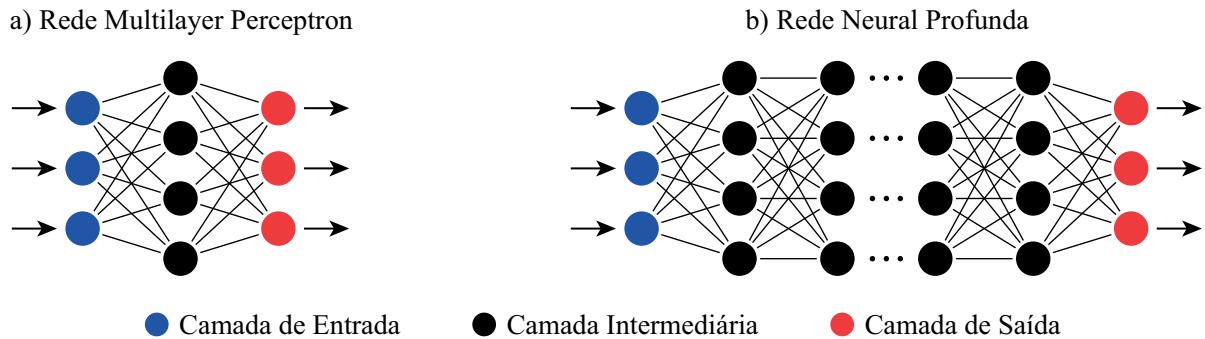


Figura 11 – Redes *Multilayer Perceptron (MLP)*. a) Rede MLP padrão. b) Rede neural profunda.

Fonte: do Autor.

A Fig. 11 mostra duas arquiteturas de redes *feedforward*, sendo a Fig. 11.a) um modelo padrão, com uma única camada intermediária e a Fig. 11.b) uma rede com várias camadas intermediárias, caracterizando uma Rede Neural Profunda (do inglês: *Deep Neural Network (DNN)*).

No geral, duas classes de redes são bastante populares e estudadas: as *redes de alimentação para frente* (do inglês: *feedforward*²) e as *redes recorrentes*, como mostradas na Fig. 12.

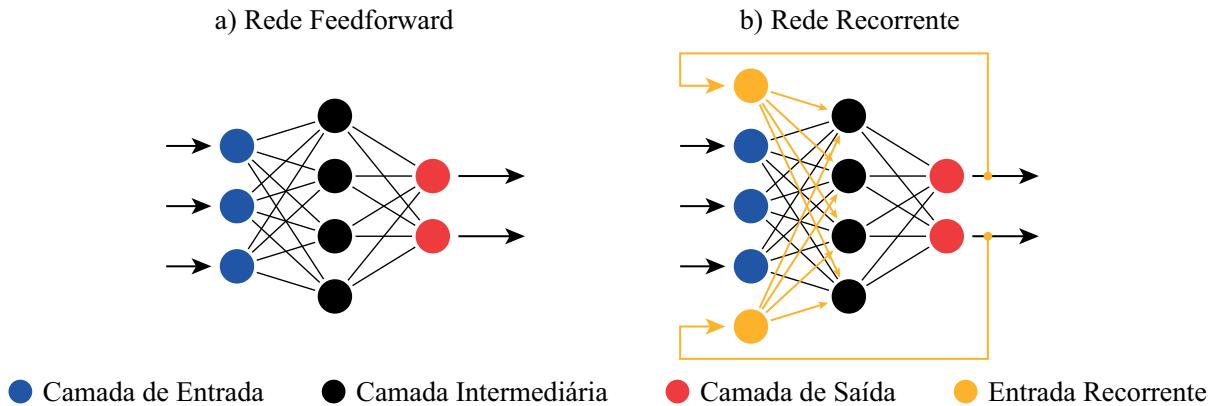


Figura 12 – Principais classes de redes neurais. a) Rede feedforward. b) Rede recorrente.

Fonte: do Autor.

Nas redes *feedforward* há um fluxo de informações unidirecional, fluindo das entradas da rede para a saída. Para as redes *recorrentes*, por outro lado, há uma realimentação da saída nas próprias entradas. Uma característica interessante das redes recorrentes é que elas podem suportar memória de curto prazo [57].

² Neste documento, será utilizado o termo *feedforward* para citar as *redes de alimentação para frente*.

2.2.6 Algoritmos e Processos de Aprendizagem

O principal objetivo de uma rede neural é aprender a partir do ambiente³ e de melhorar o seu desempenho através de um processo de aprendizagem [54]. Esse processo segue alguns passos: primeiro, a rede neural é estimulada pelo ambiente; após esse estímulo, a rede neural sofre modificações nos seus parâmetros internos (pesos sinápticos); por fim, a rede neural agora responde ao ambiente de uma forma nova, devido às modificações ocorridas na sua estrutura interna. Assim, uma rede neural irá sintetizar um modelo de aprendizado, a partir de dados do ambiente, através de um processo iterativo de ajustes dos pesos sinápticos, chamado de *treinamento* [54, 57].

Outra abordagem é como uma rede neural se relaciona com o seu ambiente, conceito esse que descreve um *paradigma de aprendizagem* [54], sendo eles *Aprendizado Supervisionado*, *Aprendizado Não Supervisionado* e *Aprendizado Por Reforço*.

No *aprendizado supervisionado*, as entradas e as saídas (banco de dados) são fornecidos por um *supervisor* (professor), como mostrado na Fig. 13. O algoritmo faz previsões iterativamente sobre os dados de treinamento e é corrigido pelo professor. O aprendizado para quando o algoritmo atinge um nível aceitável de desempenho [54].

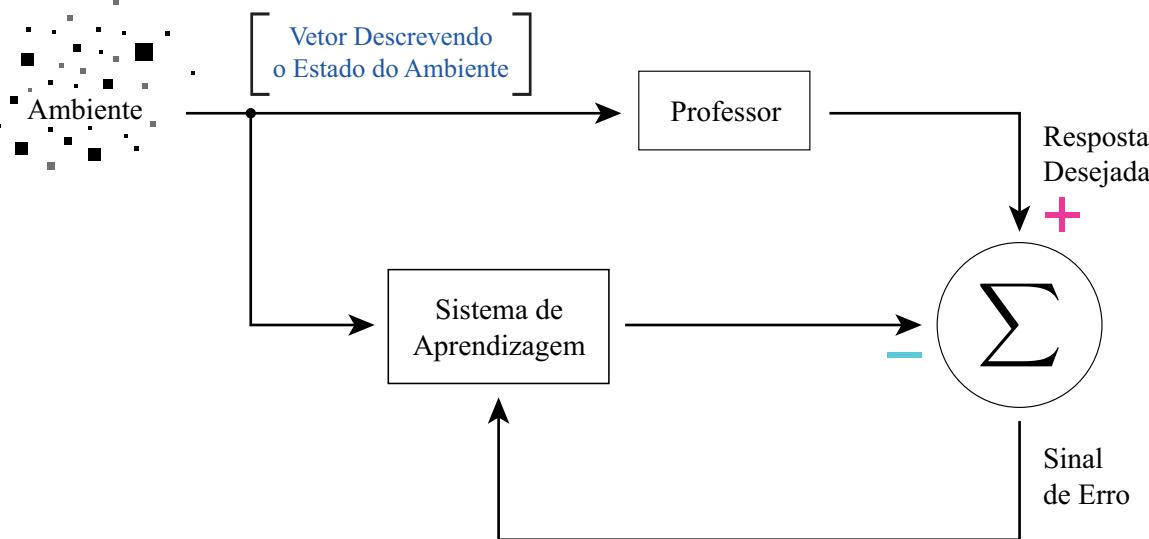


Figura 13 – Aprendizado Supervisionado.

Fonte: do Autor.

No aprendizado *não supervisionado*, a rede neural é posta ao ambiente, de forma que apenas dados de entrada são fornecidos (não há professor para acompanhar o processo de treinamento). Nesse sentido, a rede irá se auto organizar em relação às particularidades e características do conjunto de amostras que representam o ambiente [54].

³ O termo *ambiente* indica a situação para a qual uma rede neural artificial está submetida a aprender.

No *aprendizado por reforço*, a própria rede aprende a atingir um objetivo a partir do que o ambiente lhe oferece. Assim, para que a máquina faça o que o programador quer (objetivo), a inteligência artificial recebe recompensas ou penalidades pelas ações que realiza (reforço). Seu objetivo é maximizar a recompensa total [54].

Os algoritmos de aprendizado usam um conjunto de regras bem definidas, chamadas de *regras de aprendizagem*, para determinar maneira como a atualização dos pesos devem ocorrer. Os exemplos de regras mais comuns são a *regra delta*, a *regra de Hebb*, *regra perceptron* e a *regra de aprendizado competitivo*.

Muitos algoritmos de otimização (chamados de *otimizadores*) foram desenvolvidos, sendo alguns dos otimizadores mais usados são: Adagrad, Adadelta, Adam, Nadam, SGD e RMSprop. Cada um deles irá usar uma regra de aprendizagem, no contexto dos paradigmas de aprendiagem, para atualizar os pesos sinápticos de uma rede neural. Assim, os otimizadores atualizam os parâmetros de peso para minimizar a função custo por um método iterativo do *gradiente descendente* [54, 58, 59]. A função custo atua como um guia para o otimizador, dizendo se ele está se movendo na direção certa para chegar ao mínimo (local ou global), como ilustrado na Fig. 14.

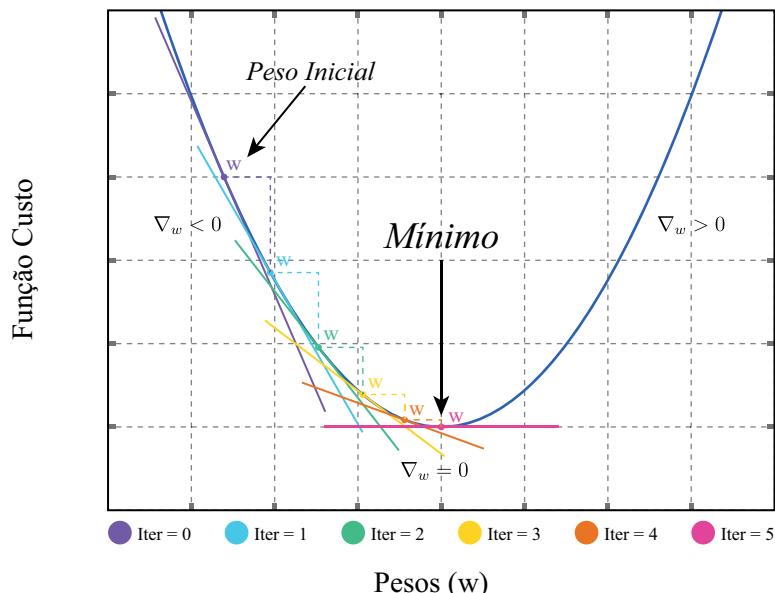


Figura 14 – Descida do gradiente.

Fonte: do Autor.

A frequência de atualização dos pesos sinápticos de uma rede neural é outro fator que deve ser avaliado. Durante o treinamento, a rede neural será exposta às instâncias do banco de dados (cada amostra do banco de dados). O tamanho do lote (*Batch*) é um hiperparâmetro que define o número de amostras a serem trabalhadas antes de atualizar os parâmetros do modelo interno. Há duas abordagens de como ocorre a correção dos pesos:

- **Modo Padrão:** a correção dos pesos sinápticos ocorre a cada apresentação de partes (batch) bem definidas do banco de dados (ver Fig. 15).
- **Modo Batch:** neste modo, apenas uma correção é feita por época. Desta maneira, o número de iterações e épocas são equivalentes (ver Fig. 15).

Tomando um exemplo prático. Um banco de dados contendo 20 instâncias (ou amostras), supondo que o tamanho de lote (*batch*) escolhido foi de 4 e o número de épocas foi de 1. Isso significa que o banco de dados será dividido em 5 lotes, cada um contendo 4 amostras. Desta forma, os pesos do modelo serão atualizados após cada lote de 4 amostras. Assim, uma época terá 5 atualizações do modelo, ou ainda, 5 iterações. Uma *época* refere-se a todas as amostras do banco de dados iteradas (pesos atualizados) uma vez, como mostra a Fig. 15.

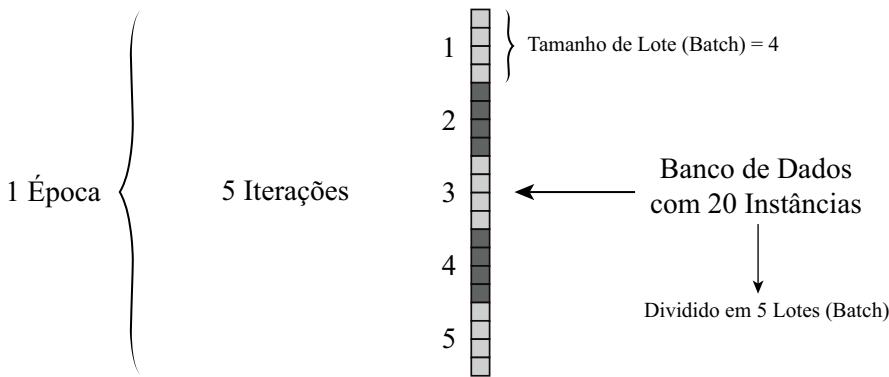


Figura 15 – Divisão das amostras do banco de dados.

Fonte: do Autor.

Com 100 épocas, o modelo será exposto por todas as amostras do banco de dados 100 vezes, processo que levará 500 iterações.

2.2.7 Algoritmo Backpropagation

O algoritmo *backpropagation* é o mais importante da história das redes neurais modernas e da aprendizagem profunda. O seu objetivo é otimizar os pesos sinápticos, através da retropropagação do erro, para que a rede neural possa aprender a mapear corretamente a relação entre os dados de entrada para com os dados de saída. O *backpropagation* é mais utilizado para o treinamento de redes *feedforward* [54], processo no qual envolve duas etapas básicas:

- **O passo para frente (*forward pass*):** nesse primeiro passo, um padrão de atividades do ambiente (vetor de entrada) é propagado através da rede, e as previsões

de saída são obtidas. Nesse contexto, os pesos sinápticos da rede são todos fixos (são iniciados de forma randômica).

- **O passo para trás (*backward pass*):** Nessa etapa, é calculado o gradiente da função custo na camada final da rede, de forma que esse gradiente é utilizado para aplicar recursivamente a regra da cadeia na função custo a fim de atualizar os pesos da rede.

Assim, o algoritmo *backpropagation* define erros no desempenho de cada neurônio, possibilitando os ajustes dos pesos sinápticos. Sendo y a saída esperada e \hat{y} a saída obtida pela rede, a função de erro (custo) é definida pelo erro médio quadrático (*Mean Squared Error (MSE)*), mostrado pela Eq. 2.9.

$$E(y, \hat{y}) = \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.9)$$

O algoritmo, então, calcula a derivada parcial da função a ser minimizada em relação ao respectivo peso, como mostrado na Eq. 2.10.

$$\frac{\partial E(y_i, \hat{y}_i)}{\partial w_{ij}} = \frac{\partial E(y_i, \hat{y}_i)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial v_j} \frac{\partial v_j}{\partial w_{ij}} \quad (2.10)$$

onde v_j é a saída do somador do neurônio j . Considerando uma função de ativação genérica $\varphi(v_j)$:

$$\frac{\partial E(y_i, \hat{y}_i)}{\partial w_{ij}} = -(y_i - \hat{y}_i) \varphi'(v_j) \frac{\partial v_j}{\partial w_{ij}} \quad (2.11)$$

onde:

$$\frac{\partial v_j}{\partial w_{ij}} = \frac{\partial(x_1w_1 + x_2w_2 + \dots + x_nw_n)}{\partial w_i} = x_n \quad (2.12)$$

portanto:

$$\frac{\partial E(y_i, \hat{y}_i)}{\partial w_{ij}} = -(y_i - \hat{y}_i) \varphi'(v_j) x_i \quad (2.13)$$

A Eq. 2.13 mostra o cálculo do gradiente da função custo. Por fim, na etapa de retropropagação, a atualização de cada um dos pesos w_{ij} pesos sinápticos é feita pela *regra delta*, definida pela Eq. 2.14.

$$w_{ij}^+ = w_{ij} - \eta \frac{\partial E(y_i, \hat{y}_i)}{\partial w_{ij}}, \quad (2.14)$$

onde w_{ij}^+ é o novo peso sináptico, w_{ij} é o peso atual e η é a taxa de aprendizagem (*learning rate*). No aprendizado de máquina, a regra delta (ver Eq. 2.15) é uma regra de aprendizado de gradiente descendente para atualizar os pesos sinápticos dos neurônios artificiais. Assim, em cada iteração, a regra de atualização dos pesos é executada, onde será definido o próximo ponto que fará a descida do gradiente, um ponto que está mais abaixo na função, rumo ao mínimo.

$$\begin{pmatrix} \text{Correção} \\ \text{de peso} \\ \Delta w(n) \end{pmatrix} = \begin{pmatrix} \text{Parâmetro da taxa} \\ \text{de aprendizagem} \\ \eta \end{pmatrix} \cdot \begin{pmatrix} \text{Gradiente} \\ \text{local} \\ \delta(n) \end{pmatrix} \cdot \begin{pmatrix} \text{Sinal de entrada} \\ \text{do neurônio j} \\ y(n) \end{pmatrix} \quad (2.15)$$

A taxa de aprendizagem é introduzida como uma constante (geralmente muito pequena), a fim de forçar o peso a ser atualizado de forma suave e lenta, por exemplo, para evitar grandes passos e comportamento caótico [54, 59]. Quando essa taxa η é muito grande, os passos de convergência são grandes também e isso pode ocasionar em *pular o mínimo* várias vezes sem nunca convergir. Quando η é muito pequeno, os passos de convergência também são pequenos, o que fará o algoritmo convergir, entretanto, poderá levar muito tempo para isso acontecer.

3 MÉTODO PROPOSTO

No Capítulo anterior, foram discutidos os conceitos básicos da *nanofotônica* e do *aprendizado de máquina*. Neste Capítulo, será abordado o procedimento de otimização e modelagem inversa por *aprendizagem profunda* dos dispositivos fotônicos estudados neste trabalho, desde a geração do banco de dados à escolha da arquitetura de rede neural. Na Seção 3.4, o leitor encontrará a metodologia de modelagem inversa aplicada aos dois circuladores baseados em cristais fotônicos.

3.1 Descrição do Problema

Em sistemas de comunicações, os dispositivos nanofotônicos desempenham um papel importante enquanto componentes não-recíprocos, como isoladores, chaves, circuladores e divisores de potência (alguns desses dispositivos são discutidos em [2–4, 60]). Uma das tarefas fundamentais nas quais esses dispositivos desempenham em circuitos integrados é na proteção de fontes eletromagnéticas contra possíveis reflexões do sinal de diferentes partes do circuito ocasionando, desta forma, na transmissão deste sinal para apenas partes desejadas do circuito [61].

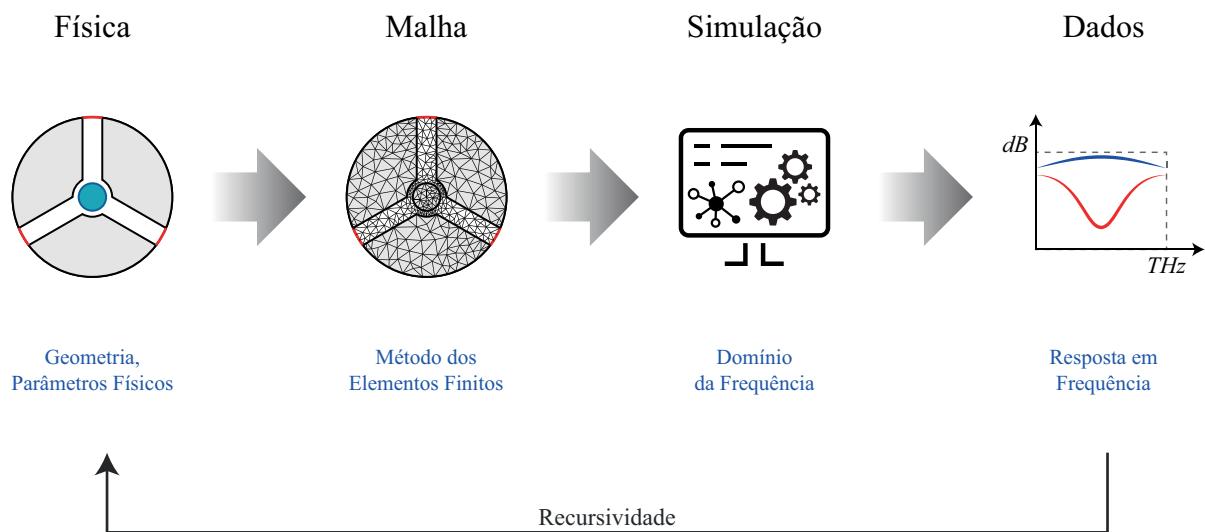


Figura 16 – Modelagem convencional no COMSOL Multiphysics ®.

Fonte: do Autor.

Esses dispositivos podem ser estudados e construídos a partir de simulações computacionais, por exemplo, por intermédio do software COMSOL Multiphysics ®, em etapas conforme mostradas na Fig. 16. Nesse sentido, esse estudo perpassa pelas seguintes etapas:

- Física: avaliação da geometria e parâmetros físicos de operação do dispositivo;
- Malha: processo de discretização do dispositivo através do Método dos Elementos Finitos;
- Simulação: início da simulação computacional no domínio da frequência.
- Dados: após finalizada a simulação, são gerados dados, como a resposta em frequência do dispositivo.

Modelar esses dispositivos através de simulação computacional é uma tarefa que pode envolver um alto custo computacional conforme aumenta a complexidade desses dispositivos [6, 7]. Em métodos tradicionais de modelagem, cabe ao projetista a recursividade para conferir manual e empiricamente as relações de geometria do dispositivo para com a resposta em frequência. Deve-se repetir esse procedimento tantas vezes quanto forem necessárias a fim de garantir uma resposta em frequência considerada ótima (deve-se avaliar enquanto parâmetro de qualidade, por exemplo, se as curvas de transmissão e isolamento estão em ressonância na frequência central de operação do dispositivo, bem como a sua largura de banda).

Por outro lado, à medida que a inteligência artificial evoluiu significativamente nas últimas décadas no campo da *aprendizagem profunda*, uma nova abordagem de modelagem de geometria de dispositivos emergiu nos últimos, como mostra a Fig. 17.

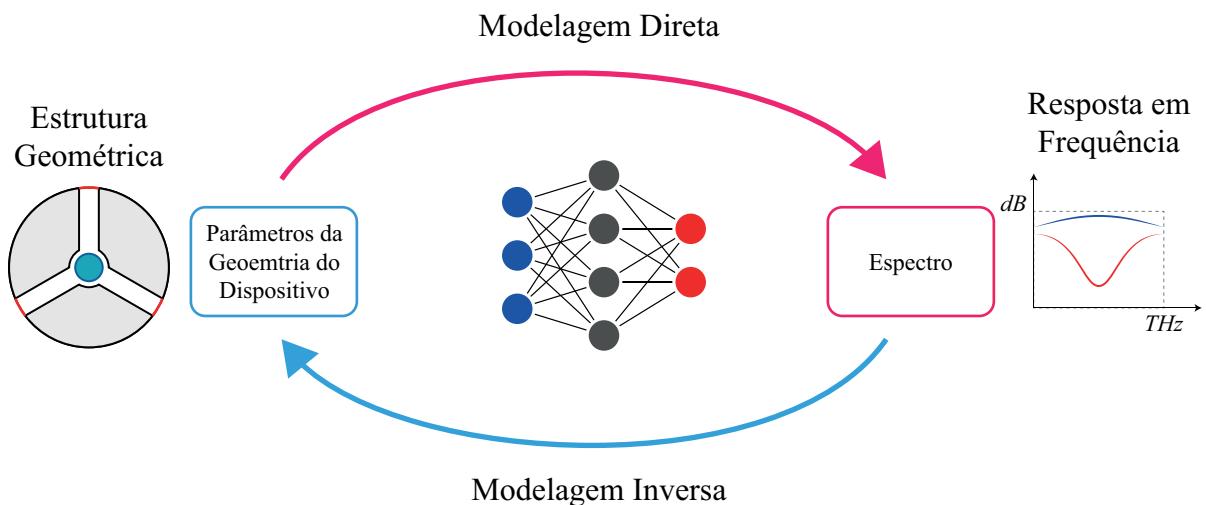


Figura 17 – Esquemático dos tipos de modelagens.

Fonte: do Autor.

A proposta é realizar a modelagem do dispositivo a partir da sua resposta em frequência, dada uma condição de operação considerada ótima ou ideal. Nesse estudo, uma rede neural profunda é usada para mapear as variadas combinações de geometria

com as respectivas respostas em frequência do dispositivo para, então, extrair um modelo de aprendizagem.

3.2 Otimização por Aprendizado Profundo

Nessa abordagem, uma rede neural profunda pode aprender como a geometria do dispositivo está relacionada com sua própria resposta em frequência. Esse processo de otimização é conhecido como *modelagem inversa* da geometria [13, 14, 16], já que é um procedimento contrário ao convencional, isto é, enquanto que na modelagem convencional contrói-se primeiramente a geometria do dispositivo para se obter a resposta em frequência a partir do seu funcionamento, neste procedimento proposto de otimização e modelagem inversa por rede neural profunda, partiremos de uma resposta em frequência ideal a qual será apresentada à rede neural já treinada, que retornará a geometria que está associada a esta resposta em frequência ideal. Resumidamente, dada uma resposta em frequência ideal (desejada), a rede neural irá modelar a geometria do dispositivo para que esse objetivo seja atingido.

Todo esse procedimento de otimização envolve as etapas seguintes etapas básicas:

1. **Estudo Inicial:** nesta etapa, são verificadas as características dos dispositivos, como a sua modelagem geométrica e resposta em frequência. Esse estudo é importante para avaliar quais parâmetros devem ser levados em consideração para a otimização do dispositivo.
2. **Construção do Banco de Dados:** consiste em montar um banco de dados contendo as características de geometria de dispositivo relacionadas com a respectiva resposta em frequência. Esse procedimento foi realizado usando os softwares COM-SOL e o MATLAB.
3. **Rede Neural Profunda:** a rede neural foi construída na linguagem de programação Python com o uso do framework TensorFlow. O objetivo desse procedimento é extrair um modelo de aprendizagem para tornar as previsões com maior acurácia. No final desse processo, é esperável que a rede neural (*deep learning*) aprenda qual é a geometria mais otimizada para os dispositivos estudados.
4. **Algoritmo de Otimização:** todo o procedimento de otimização envolve vários serviços e programas e, a fim de automatizar todo esse processo, foi desenvolvido também um script que concatena todos as etapas usadas para a otimização. O objetivo é esse algoritmo de otimização rodar até que o modelo otimizado seja encontrado.

3.2.1 Construção do Banco de Dados

O procedimento de otimização por inteligência artificial envolve, primeiramente, coletar os dados do problema e organizá-los em um banco de dados. Estes dados foram coletados por meio da API (*Application Programming Interface*, ou ainda em português, Interface de Programação de Aplicativos) que o próprio COMSOL oferece. Uma API permite que aconteça troca de informações entre dois ou mais sistemas. Neste caso, a API *COMSOL LiveLink for MATLAB* [62] permite que o MATLAB possa controlar como as simulações numéricas do COMSOL acontecem, desde o processo de automatizá-las à triagem desses dados para a construção do banco de dados.

Os dispositivos nanofotônicos são modelados e estudados a partir do software de simulação eletromagnética COMSOL. Nessa etapa, o próprio projetista constrói o dispositivo com as características geométricas e de operação que ele deve ter. Uma vez que esse estudo esteja estabelecido, o próximo passo é mapear variáveis que mais influenciam na resposta em frequência do dispositivo. Assim, cada variável será responsável por realizar uma determinada modificação na geometria do dispositivo. Nesse ponto, deve-se atentar que essas variáveis geométricas devem respeitar a condição de simetria dos dispositivos. Isso é necessário pois é realizado um estudo embasado na *teoria de grupos* [63], a qual permite reduzir a quantidade de cálculos da matriz de espalhamento do dispositivo.

As mesmas variáveis que modificam simetricamente a geometria do dispositivo são carregadas em um *script* no MATLAB. A ideia por trás desse script é automatizar as simulações em *Loops*¹ de execução. Desta maneira, a cada execução do script o MATLAB atribui valores randômicos para essas variáveis. Em termos práticos, isso significa que cada simulação de dispositivo haverá uma geometria diferente para simular. Ao final de cada simulação, os dados randômicos de geometria e a respectiva resposta em frequência são exportados em vários arquivos com a extensão *.txt* e armazenados em um diretório definido no mesmo script do MATLAB.

Foi desenvolvido, também, um outro script no MATLAB que é responsável por ler os dados gerados pelas simulações automatizadas. A função desse *script* é ler todos os arquivos de simulação que foram gerados e montar o banco de dados em um único arquivo, onde os dados estão organizados de maneira sequencial (conforme o Loop de simulação) e normalizados no intervalo 0 . . . 1, como mostrado na Fig. 18.

¹ Neste trabalho, o termo *Loop* é usado para designar um conjunto de várias simulações.

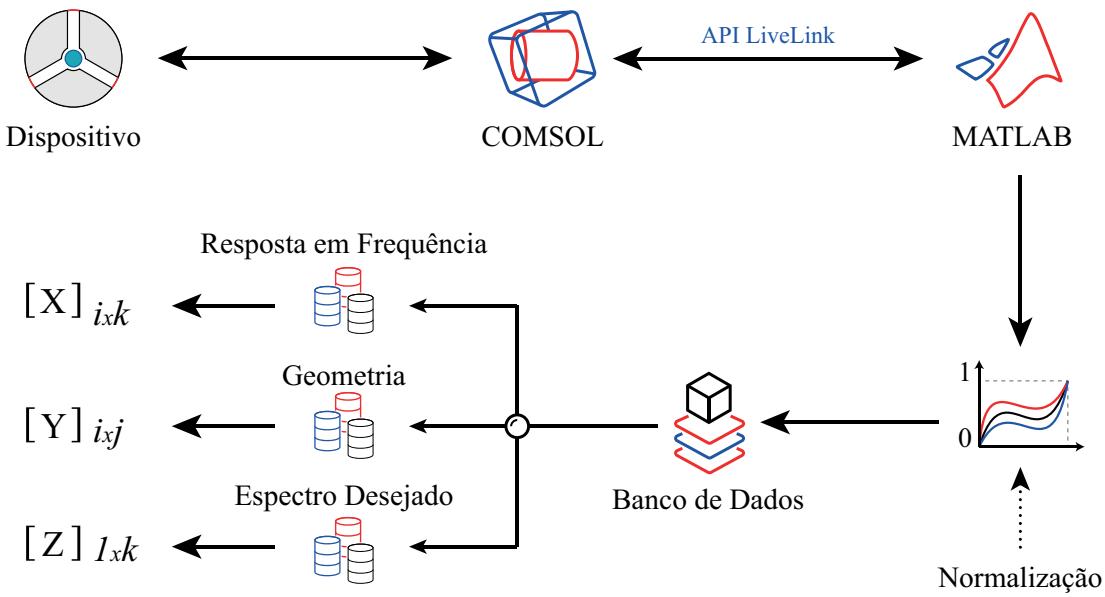


Figura 18 – Procedimento de contrução do banco de dados.

Fonte: do Autor.

Assim, nessa etapa, são gerados três arquivos na extensão `.csv`, cada qual entendidos como tensores, a saber:

- **Resposta em Frequência** – definido pelo tensor $[X]$, compreende as amplitudes discretizadas em 51 pontos de gráfico de cada curva da resposta em frequência, sequenciadas linha-a-linha.
- **Geometria** – definido pelo tensor $[Y]$, compreende o agrupamento de cada valor randômico de parâmetros de geometria.
- **Espectro Desejado** – definido pelo tensor $[Z]$, compreende à resposta em frequência desejada para o dispositivo (construída manualmente no MATLAB).

Esses arquivos podem ser visualizados na forma de tensores que alimentarão a rede neural profunda. A Eq. 3.1 mostra o tensor $[Y]$ referente aos dados de geometria do dispositivo.

$$[Y]_{i,j} = \begin{bmatrix} y_{1_{[1,1]}} & y_{2_{[1,2]}} & \cdots & y_{j_{[1,j]}} \\ y_{1_{[2,1]}} & y_{2_{[2,2]}} & \cdots & y_{j_{[2,j]}} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1_{[i,1]}} & y_{2_{[i,2]}} & \cdots & y_{j_{[i,j]}} \end{bmatrix}, \quad (3.1)$$

onde j é o número total de variáveis que modificam a geometria do dispositivo e i é referente ao número de instâncias, isto é, ao número de simulações totais. O mesmo

raciocínio se aplica para a construção do banco de dados da resposta em frequência, a qual é agrupada em um tensor como mostrado na Eq. 3.2:

$$[\mathbf{X}]_{i,k} = \begin{bmatrix} x_{1[1,1]} & x_{2[1,2]} & \cdots & x_{k[1,k]} \\ x_{1[2,1]} & x_{2[2,2]} & \cdots & x_{k[1,k]} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1[i,1]} & x_{2[i,2]} & \cdots & x_{k[i,k]} \end{bmatrix}. \quad (3.2)$$

Note que o número de instâncias i precisa ter o mesmo tamanho para $[\mathbf{X}]$ e $[\mathbf{Y}]$, uma vez que geometria e resposta em frequência estão relacionadas. Por fim, o *espectro desejado*² é definido como um tensor, assim mostrado na Eq. 3.3.

$$[\mathbf{Z}]_{1,k} = [z_{1[1,1]} \ z_{2[1,2]} \ \cdots \ z_{k[1,k]}]. \quad (3.3)$$

Como já mencionado, o tensor $[\mathbf{Z}]$ foi feito considerando as características ideais de operação do dispositivo em sua resposta em frequência. É importante ressaltar que a dimensão k dos tensores $[\mathbf{X}]$ e $[\mathbf{Z}]$ tem o mesmo tamanho (número de colunas). Nesse sentido, k é definido dado o número de curvas da resposta em frequência (parâmetro que depende de cada dispositivo avaliado) e o quão essas curvas estão discretizadas (previamente comentado, em 51 pontos). Essa análise está melhor elucidada na Seção 3.4 deste Capítulo.

Todo o banco de dados é normalizado no intervalo $0 \dots 1$, conforme a Eqs. 3.4, 3.5 e 3.6:

$$N_{[Y]} = \frac{DataSet[Y] - MinDataSet[Y]}{MaxDataSet[Y] - MinDataSet[Y]}, \quad (3.4)$$

$$N_{[X]} = \frac{DataSet[X] - MinDataSet[X]}{MaxDataSet[X] - MinDataSet[X]}, \quad (3.5)$$

$$N_{[Z]} = \frac{DataSet[Z] - MinDataSet[Y]}{MaxDataSet[Y] - MinDataSet[Y]}, \quad (3.6)$$

onde $N_{[Y]}$, $N_{[X]}$ e $N_{[Z]}$ são as respectivas normalizações dos tensores $[\mathbf{Y}]$, $[\mathbf{X}]$ e $[\mathbf{Z}]$. Esse procedimento é necessário, pois a rede neural irá usar um método recursivo para minimizar uma *função custo* usando a descida do *gradiente descendente* [58, 59] e, desta forma, reduzir a influência de valores grandes no banco de dados em relação aos valores pequenos, de forma a redimensionar o banco de dados sem distorcer as diferenças nos intervalos entre

² O termo *espectro desejado* tem o mesmo significado de *resposta em frequência desejada*.

cada valor. Dados muito dispersos (portanto, alto desvio padrão) podem tornar o algoritmo de otimização da rede neural mais lento, ou até mesmo provocar a não-convergência da descida do gradiente [59].

3.2.2 Rede Neural Profunda

A rede neural tem por objetivo extrair um modelo de aprendizagem a partir do banco de dados previamente montado, como ilustrado na Fig. 19.

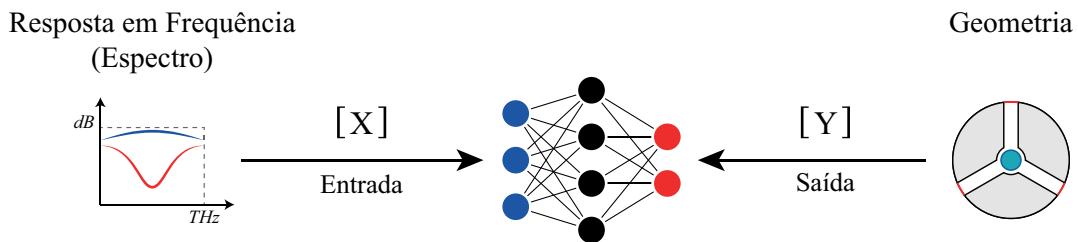


Figura 19 – Esquema básico da alimentação do banco de dados na rede neural.

Fonte: do Autor.

Para a construção da rede neural profunda foi utilizada a biblioteca de código aberto para Aprendizado de Máquina *Tensorflow* [64] no ambiente da linguagem de programação Python [65].

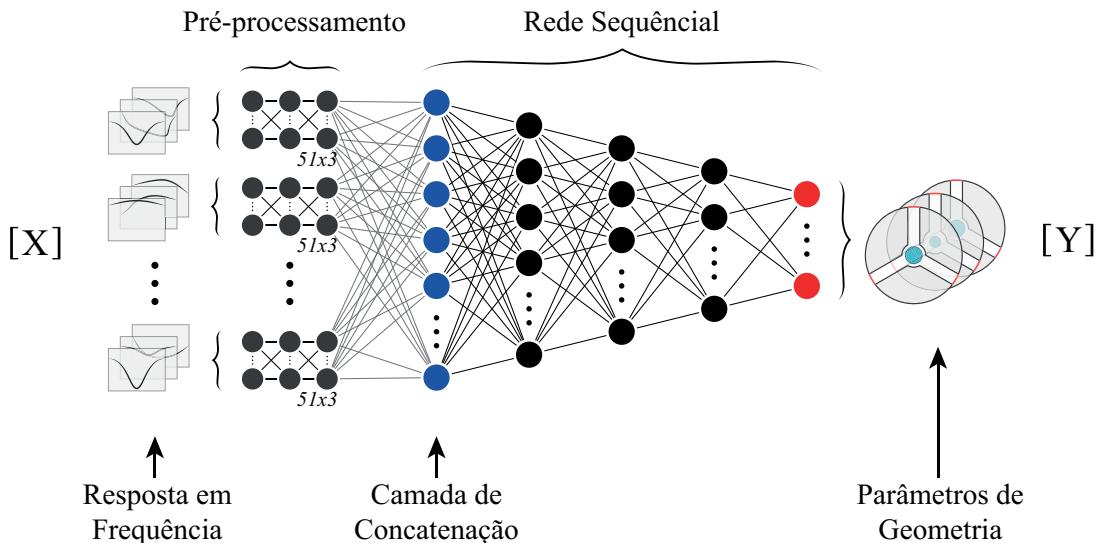


Figura 20 – Arquitetura da rede neural profunda detalhada.

Fonte: do Autor.

A Fig. 20 mostra a arquitetura detalhada da rede neural e o esquemático de como o banco de dados é alimentado através dela. O tensor $[X]$ é alimentado através da entrada da rede, passando por uma camada de *pré-processamento paralelo*, desta forma,

cada estrutura em paralelo cuidará de fazer um pré-processamento de cada curva individualmente. Posteriormente, os dados seguem para uma camada de concatenação e, então, propagados adiante através da *rede sequencial* onde, na saída de toda a rede, estará o tensor de parâmetros de geometria $[Y]$.

A rede neural, primeiramente, passa pelo processo de treinamento, onde ela aprenderá iterativamente a modelagem das relações da geometria do dispositivo com a sua resposta em frequência. Posteriormente, a rede neural passará pelo processo de predição, isto é, após ela ter extraído um modelo de aprendizagem do dispositivo, agora irá predizer qual geometria deve-se ter, dada uma resposta em frequência ideal.

3.2.3 Treinamento e Predição

No processo de treinamento da rede neural profunda, foi utilizado o algoritmo de aprendizagem *Adam* e, como função custo, o erro médio quadrático (do inglês, *mean squared error* (MSE)). O processo de treinamento é feito minimizando a função custo, C , mostrada na Eq. 3.7.

$$C = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2. \quad (3.7)$$

A partir das métricas do MSE, pode-se estimar a performance da rede neural profunda. Na Eq. 3.7, o parâmetro i é o mesmo *número de instâncias* anteriormente mencionado, n é o número de execuções de simulações (*Loop*) e os argumentos Y_i e \hat{Y}_i são os vetores de *geometria real* e *geometria prevista*, respectivamente. Nesse aspecto, a função custo C mede o erro entre a *geometria real* Y_i e *geometria prevista* \hat{Y}_i . Desse modo, o caso ideal é quando a função custo C for o mais próxima quanto possível de zero. Em termos práticos, isso indica que não há mais erro significativo entre a predição e o valor real da geometria e, assim, a rede neural se tornou apta a predizer qual é a geometria que está relacionada à resposta em frequência desejada.

Após o processo de treinamento, a rede neural poderá fazer a predição da geometria dada um espectro desejado. Os dados da saída da rede contém um vetor (ainda normalizado) com as variáveis de geometria que deverão, posteriormente, ser simuladas no COMSOL para a validação da predição da rede neural. É necessário *desnormalizar* esse vetor para a devida simulação.

$$D_{[Y]} = Net[W] \times (MaxDataSet[Y] - MinDataSet[Y]) + MinDataSet[Y] \quad (3.8)$$

Na Eq. 3.8, $Net[W]$ refere-se ao vetor de saída da rede na etapa de predição (ainda normalizado), e $D_{[Y]}$ é o vetor de geometria desnormalizado. Nesse processo, a desnormalização será feita usando os mesmos valores de máximo e mínimo da Eq. 3.4.

O banco de dados foi dividido em 80% para dados de *treinamento*, 10% para *validação* e 10% para *teste*. Em termos práticos, isso significa que a rede neural terá apenas 80% para treinar as amostras de geometria e resposta em frequência. Posteriormente, durante a fase de teste, a rede neural será submetida a 10% do banco de dados, sobre o qual ela não teve conhecimento prévio (ou seja, não foi enviesada pela amostra). E é neste momento que o poder de predição será avaliado.

3.2.4 Procedimento de Otimização

Muitos serviços são executados em etapas, desde o momento em que o banco de dados é extraído e montado, ao momento de validação da predição da rede neural. Assim, foi desenvolvido um script que automatiza a comunicação entre esses serviços (COMSOL, MATLAB e PYTHON/TENSORFLOW).

Todos eles são integrados dinamicamente por meio de um algoritmo executado a partir do terminal do computador. Assim, como ilustrado na Fig. 21, o algoritmo pode ser executado automaticamente quantas vezes forem necessárias (até que a otimização seja concluída, por exemplo).

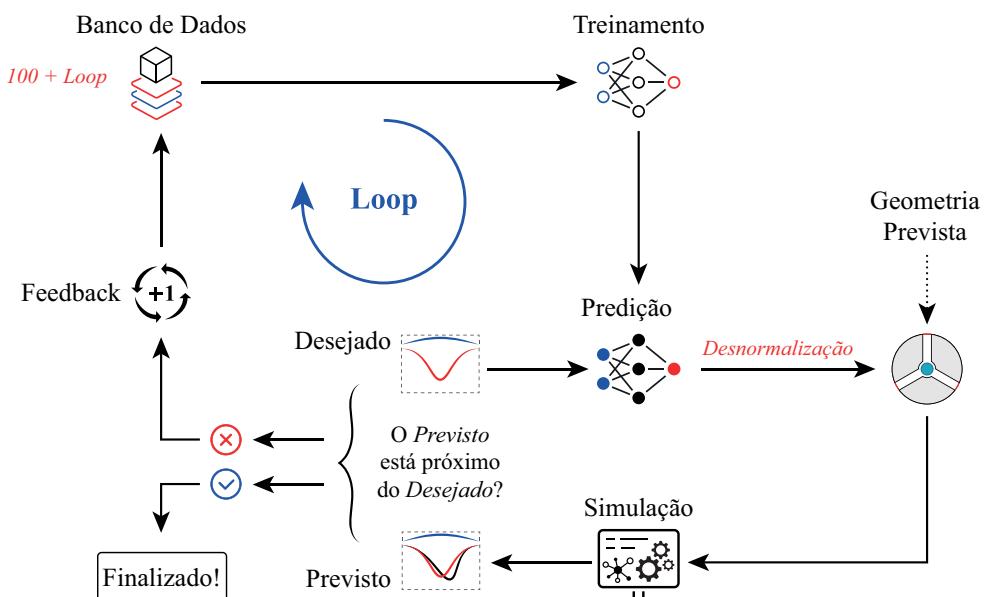


Figura 21 – Algoritmo de otimização.

Fonte: do Autor.

Como ponto de partida, foi gerado um *Loop* de 100 simulações aleatórias, contendo as mais diversas variações de geometria e respectiva resposta em frequência, a fim de construir um banco de dados inicial, denotado por i_0 na Fig. 22.

Esse número inicial (100 instâncias iniciais no banco de dados) é necessário para

se averiguar a melhor arquitetura de rede que se adapta a cada problema. Nesse ponto, várias arquiteturas de redes neurais com diferentes configurações de neurônios e camadas intermediárias serão avaliadas através das métricas de desempenho do erro médio quadrático com a finalidade de se trabalhar com a melhor rede neural. Superando essa necessidade da arquitetura de rede, os *Loops* de simulações continuam a partir de i_0 , como mostrado na Fig. 22.

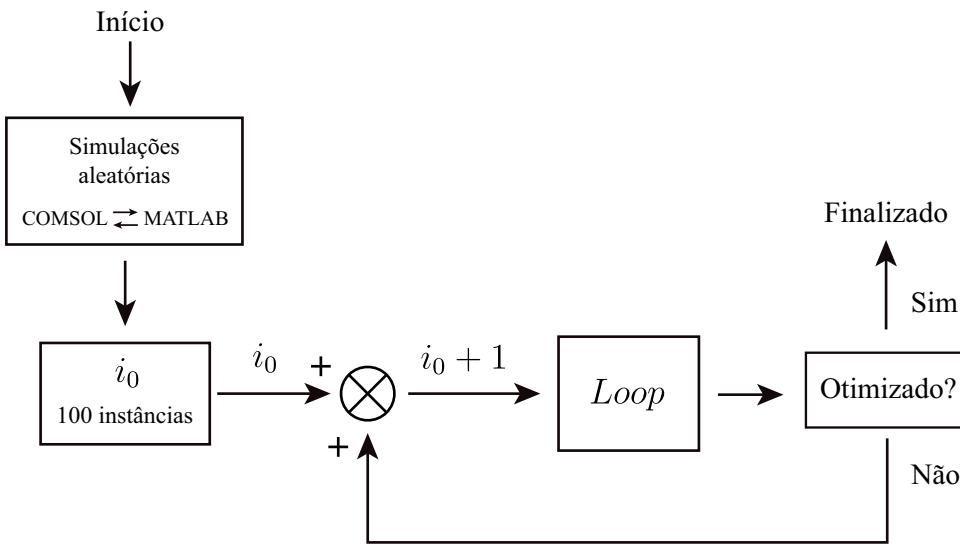


Figura 22 – Diagrama de blocos simplificado do algoritmo de otimização.

Fonte: do Autor.

Vale ressaltar que cada dispositivo submetido ao procedimento de modelagem inversa tem o seu próprio banco de dados e, portanto, o procedimento de otimização funciona de forma totalmente independente para cada um.

Seguindo com o algoritmo, a rede neural profunda será treinada com o banco de dados por 20000 épocas. Posteriormente, na fase de *predição*, será apresentado o tensor $[Z]$ contendo o espectro desejado. A rede neural, então, retorna os parâmetros de geometria que serão desnormalizados (ver Eq. 3.8) e, posteriormente, simulados no COMSOL via *API LiveLink for MATLAB*. A resposta em frequência resultante da simulação é então comparada com o *espectro desejado*. Nesse momento, o algoritmo toma uma decisão: se a resposta em frequência *prevista* (simulada) está satisfatoriamente próximo ao *desejado*, o procedimento de otimização é finalizado. Isso significa que a geometria do dispositivo com a resposta de frequência contendo parâmetros ótimos de operação foi encontrada. Porém, se o resultado da simulação não for satisfatório, esse resultado não é descartado, mas sim realimentado no banco de dados na fase de *feedback*. Como consequência, isso incrementa mais uma instância ($i + 1$) no banco de dados. Dessa forma, da próxima vez que o algoritmo for executado, a rede neural saberá que essa ainda não é a solução.

3.3 Fatores de Qualidade

Foram realizadas algumas avaliações de fatores que aferem a qualidade dos dispositivos estudados. Este estudo é necessário para se averiguar as possíveis melhorias em termos de desempenho dos dispositivos proporcionados pelo método de otimização por aprendizado de máquina. Esses *fatores de qualidade* são características da resposta em frequência inerente a cada dispositivo estão descritos como se segue:

- **Frequência central:** todas as curvas dos parâmetros-S (S_{ij}) são avaliadas quanto à ressonância na frequência central de operação do dispositivo. Esse fator é avaliado por ΔF_{ij} , que afere a distância do ponto de inflexão da curva avaliada à F_c (frequência central). Um fator ótimo ou ideal é quando $\Delta F_{ij} = F_c$.
- **Transmissão:** as curvas relacionadas à transmissão são avaliadas quanto à proximidade de -1 dB. Esse fator é avaliado por $\Delta T1_{ij}$. Um valor considerado ótimo é quando $\Delta T1_{ij} = -1$ dB.
- **Isolamento e reflexão:** as curvas relacionadas ao *isolamento* e *reflexão* são avaliadas quanto à proximidade do limiar de -20 dB. Esse fator é avaliado por $\Delta T2_{ij}$. Um valor considerado ótimo é quando $\Delta T2_{ij} \leq -20$ dB.
- **Largura de banda:** a largura de banda do dispositivo é avaliada a -15 dB. Esse fator é avaliado por $BW = f_2 - f_1$, onde f_1 e f_2 são referentes às curvas mais internas (mais próximas à F_c). Um valor considerado ótimo é quando $BW \gg 0$.

A Fig. 23 ilustra como esses fatores estão relacionados na resposta em frequência.

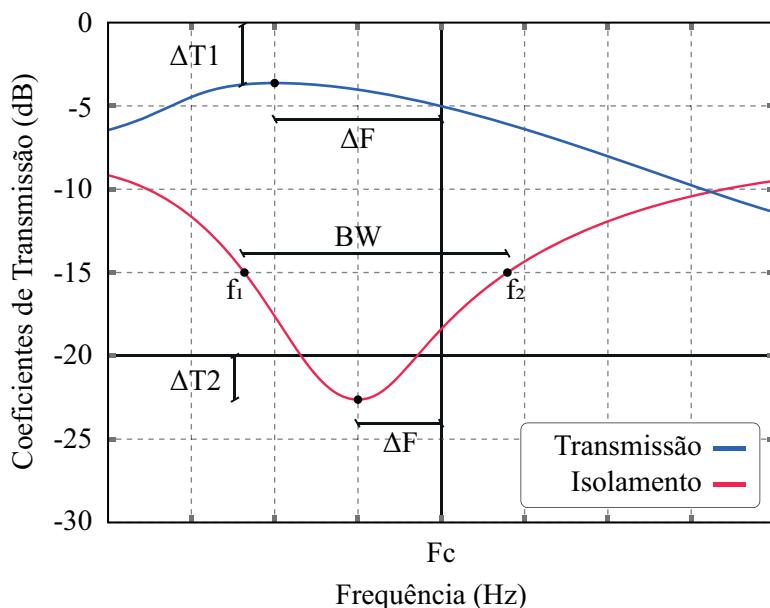


Figura 23 – Fatores de qualidade avaliados na resposta em frequência.

Fonte: do Autor.

3.4 Aplicação em Circulador

Os circuladores de junção-T baseados em cristal fotônicos, sendo um com modo de ressonância *dipolo* e outro com modo de ressonância *quadrupolo* discutidos em [17], foram submetidos ao procedimento de otimização e modelagem inversa proposto neste trabalho, e apresentam a geometria base indicada na Fig. 24.

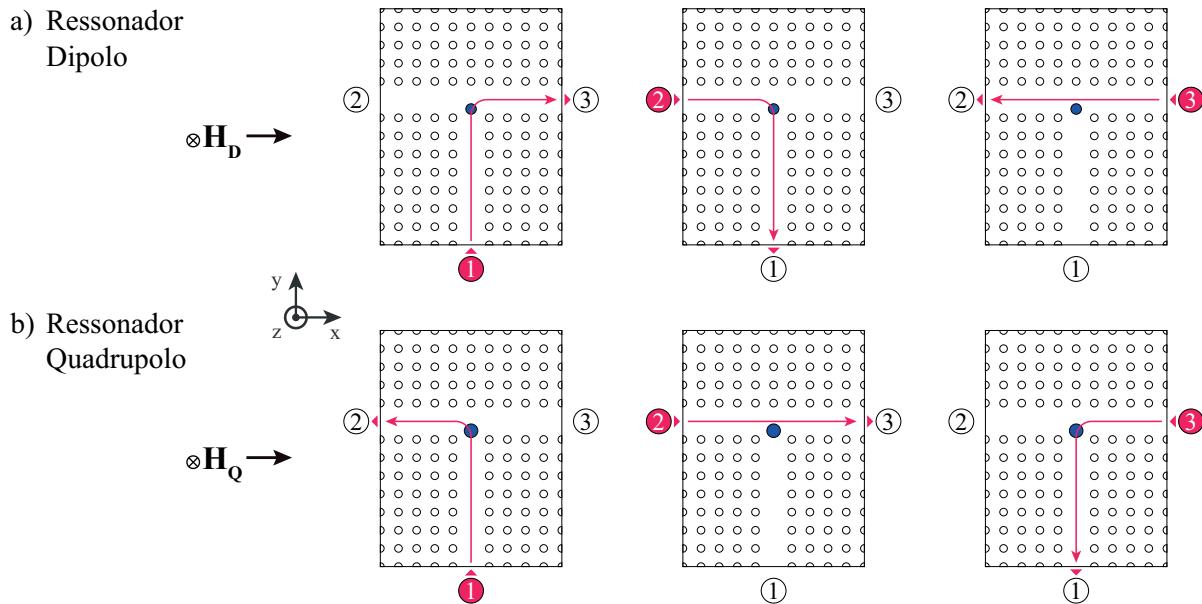


Figura 24 – Funcionamento dos circuladores. a) Modo dipolo. b) Modo quadrupolo

Fonte: do Autor.

A resposta em frequência é obtida considerando o desempenho do sinal da *porta 1* para todas as outras e vice-versa. Portanto, 9 curvas são avaliadas:

- Porta 1: S_{11} , S_{21} , S_{31}
- Porta 2: S_{12} , S_{22} , S_{32}
- Porta 3: S_{13} , S_{23} , S_{33}

Conforme a Fig. 24, para o circulador *dipolo*, as curvas S_{31} , S_{12} e S_{23} devem ser maximizadas (sinal propagado) e as curvas S_{11} , S_{21} , S_{22} , S_{32} , S_{13} e S_{33} minimizadas (sinal isolado). A mesma análise se estende para o circulador *quadrupolo*, onde as curvas S_{21} , S_{32} e S_{13} devem ser maximizadas (sinal propagado) e as curvas S_{11} , S_{31} , S_{12} , S_{22} , S_{23} e S_{33} minimizadas (sinal isolado).

A parametrização da geometria de ambos circuladores foi realizada considerando uma área que, quando modificadas, implicam nas maiores contribuições para o compor-

tamento da resposta em frequência de cada dispositivo. Essa *área de modelagem* está mostrada na Fig. 25 e indica a área central da *junção-T*.

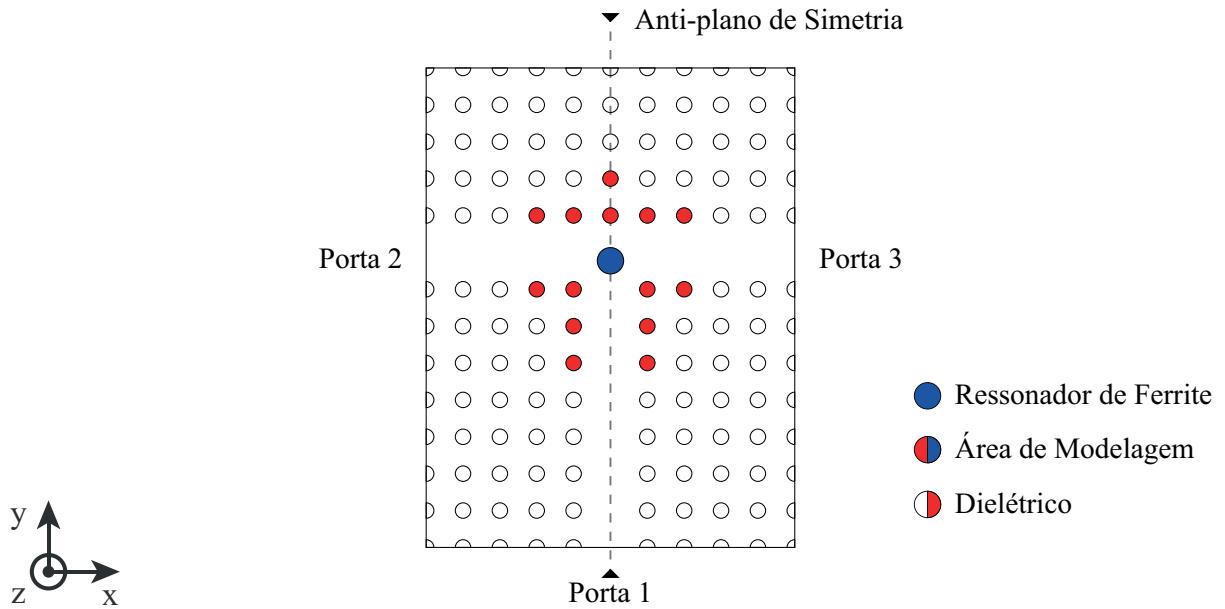


Figura 25 – Geometria do cristal fotônico.

Fonte: do Autor.

Essas variáveis de geometria incluem os deslocamentos no *eixo-x*, *eixo-y* e o *raio* de cada cilindro, respeitando o eixo de simetria do dispositivo. No total, foram 24 variáveis de geometria para ambos dispositivos, portanto, o tensor $[Y]$ (alimentado na saída da rede) terá dimensão $i \times 24$ para ambos dispositivos.

Cada curva da resposta em frequência de ambos circuladores está discretizada em 51 pontos e, como em cada porta são avaliadas 3 curvas, portanto, $51 \times 3 \times 3 = 459$. Isso indica que os tensores $[X]$ e $[Z]$ terão dimensão $i \times 459$ e 1×459 , respectivamente. Nesse contexto, a rede neural profunda terá 459 neurônios na camada de entrada e 24 neurônios na camada de saída para ambos circuladores.

3.4.1 Arquitetura de Rede

A arquitetura da rede foi definida a partir da quantidade de neurônios nas camadas de entrada e saída. Foram desenvolvidas 6 arquiteturas de redes com diversas configurações de neurônios e camadas.

Definida a quantidade de neurônios nas camadas de entrada e saída da rede neural profunda, o próximo passo foi definir as camadas intermediárias (quantidade de camadas e quantidade de neurônios em cada camada). A escolha igual das mesmas quantidades de variáveis de geometria para ambos circuladores foi intencional, pois facilita o uso de

uma arquitetura para trabalhar ambos dispositivos. Foram desenvolvidas 6 arquiteturas de redes com diversas configurações de neurônios e camadas, a saber:

- Rede 1: $459 - 200 - 24$
- Rede 2: $459 - 200 - 100 - 24$
- Rede 3: $459 - 300 - 200 - 100 - 24$
- Rede 4: $(51) \parallel \times 9 - 459 - 300 - 200 - 100 - 24$
- Rede 5: $(51 - 51 - 51) \parallel \times 9 - 459 - 300 - 200 - 100 - 24$
- Rede 6: $(51 - 51 - 51 - 51 - 51 - 51) \parallel \times 9 - 459 - 300 - 200 - 100 - 24$

As 6 arquiteturas de redes foram avaliadas com o banco de dados inicial i_0 . Primeiramente, foi realizado o teste de performance das três primeiras redes (*rede 1*, *rede 2* e *rede 3*). Foi observado que a *rede 3* obteve o melhor desempenho (isto é, menor erro). Após essa etapa, mais três arquiteturas foram geradas (*rede 4*, *rede 5* e *rede 6*). Essas redes são variações da *rede 3* contendo nove redes de pré-processamento paralelo (uma para cada curva avaliada), variando em 1, 3 e 6 camadas, as quais são conectadas posteriormente à rede sequencial (propriamente dita, a *rede 3*) através da camada de concatenação.

Tabela 1 – Performance de cada arquitetura de rede em relação ao banco de dados inicial i_0 de cada circulador.

Rede	Circulador Dipolo (MSE)	Circulador Quadrupolo (MSE)
1	$3,6307e^{-2}$	$4,3351e^{-2}$
2	$2,8347e^{-2}$	$3,7666e^{-2}$
3	$2,6205e^{-2}$	$3,5608e^{-2}$
4	$2,5085e^{-2}$	$3,3075e^{-2}$
5	$2,4539e^{-2}$	$3,2202e^{-2}$
6	$2,5934e^{-2}$	$3,3595e^{-2}$

Fonte: do Autor.

A Tabela 1 mostra o desempenho de cada arquitetura de rede para com o banco de dados inicial i_0 . A arquitetura de *rede 5* obteve o menor erro (MSE) para o banco de dados de ambos circuladores e, portanto, foi a escolhida.

3.4.2 Operação Ideal

Foi desenvolvida uma *resposta em frequência desejada* para ambos circuladores, como mostrada na Fig. 26.

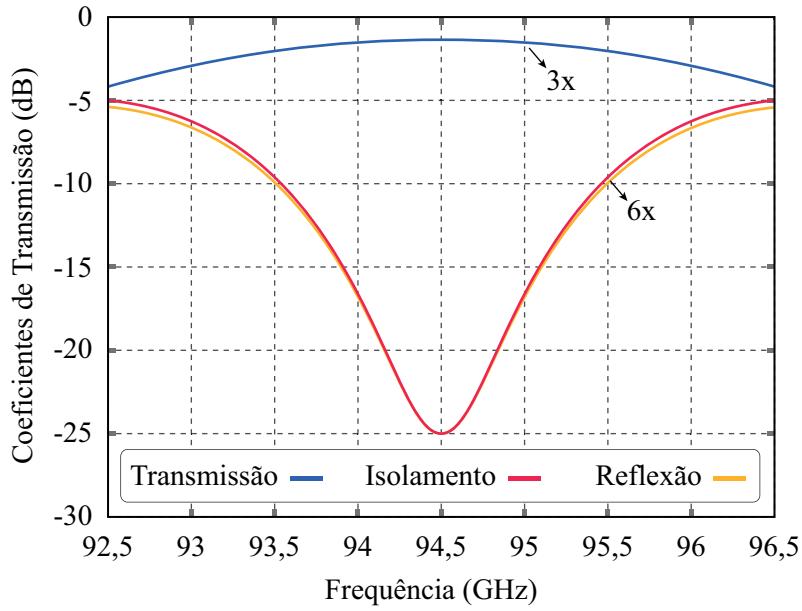


Figura 26 – Resposta em frequência desejada.

Fonte: do Autor.

Ambos dispositivos foram projetados para operar na frequência central de 94,5 GHz, portanto, todas as curvas estão em ressonância a essa frequência. As curvas relacionadas ao *isolamento* e a *reflexão* têm os seus coeficientes de transmissão com vale em -25 dB.

É válido ressaltar que essas características de operação idealizadas na Fig. 26 foram definidas às proximidades de resposta em frequência de simulações já observadas dos referidos dispositivos. Construir uma resposta em frequência desejada muito distante do que o dispositivo habitualmente se comporta, pode levar o modelo em redes neurais a nunca chegar em uma solução. Seria o caso, por exemplo, se as curvas de isolamento e reflexão estivessem com o vale a -200 dB. Certamente, essa poderia ser uma solução intangível no *espaço de design*.

4 RESULTADOS

No Capítulo anterior, foi visto o procedimento metodológico de otimização e modelagem inversa por aprendizagem profunda. Neste Capítulo, são mostrados os resultados do procedimento para os dispositivos estudados. Na Seção 4.1, são mostrados os resultados referentes aos dois circuladores baseados em cristais fotônicos.

4.1 Circulador

Os resultados da modelagem inversa dos dois circuladores de junção-T baseados em cristal fotônico estão apresentados nesta Seção. A Fig. 27 mostra a resposta em frequência do circulador de ressonância *dipolo* após o procedimento de otimização proposto neste trabalho.

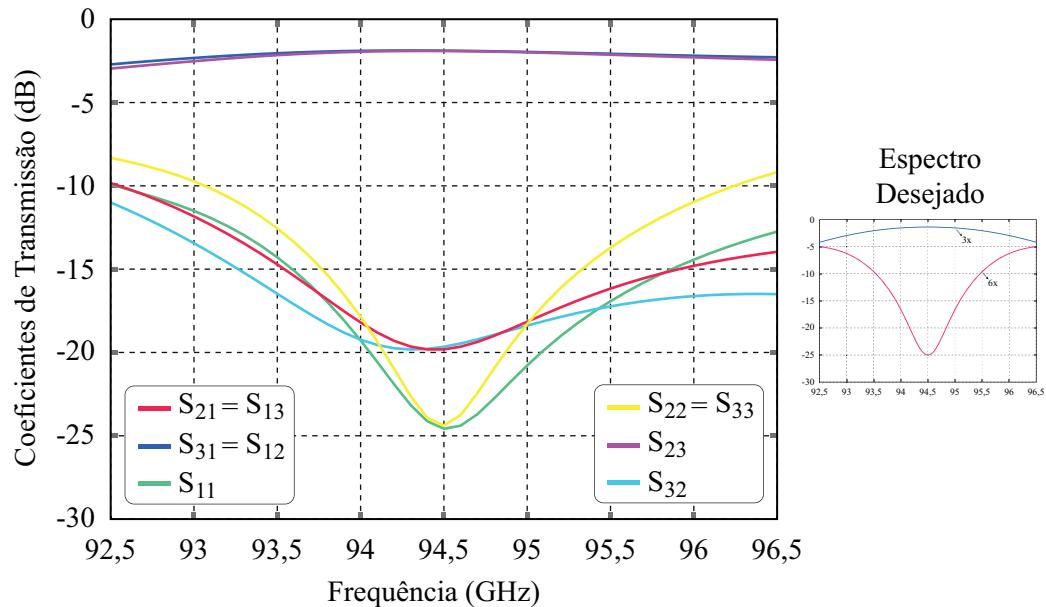


Figura 27 – Resposta em frequência do circulador dipolo após a modelagem inversa.

Fonte: do Autor.

Nota-se que o procedimento de modelagem inversa foi satisfatório e é possível notar a proximidade com a resposta em frequência desejada (ver Fig. 26 para mais detalhes). O resultado do circulador com ressonância *quadrupolo* também apresenta êxito na aproximação com a resposta em frequência desejada, como mostra a Fig. 28.

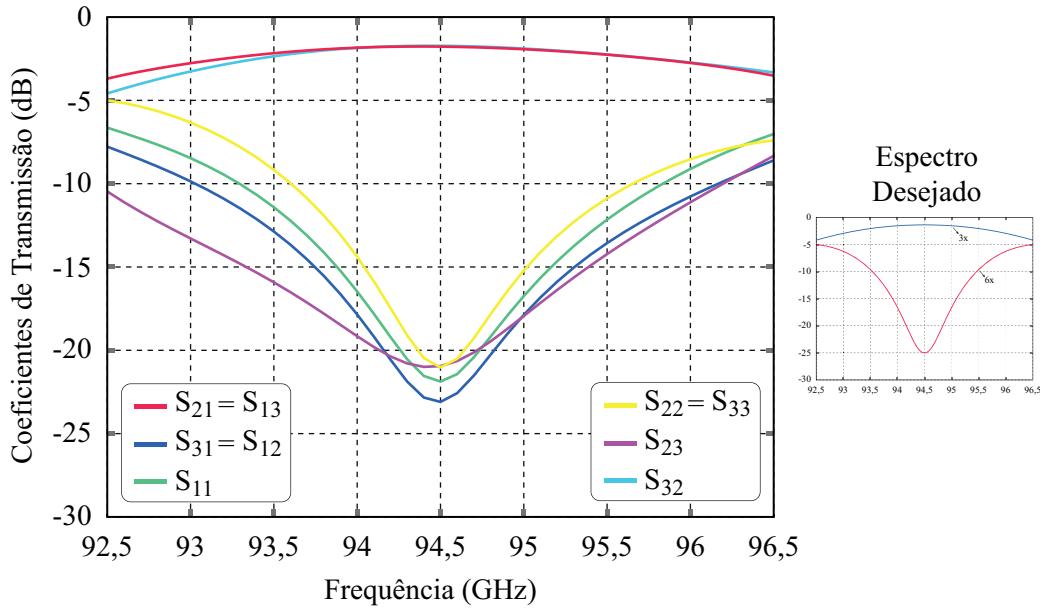


Figura 28 – Resposta em frequência do circulador quadrupolo após a modelagem inversa.

Fonte: do Autor.

A geometria final de ambos circuladores está mostrada na Fig. 29. Cada variável geométrica otimizada está abordada detalhadamente na Seção 6.2 do Apêndice 6.

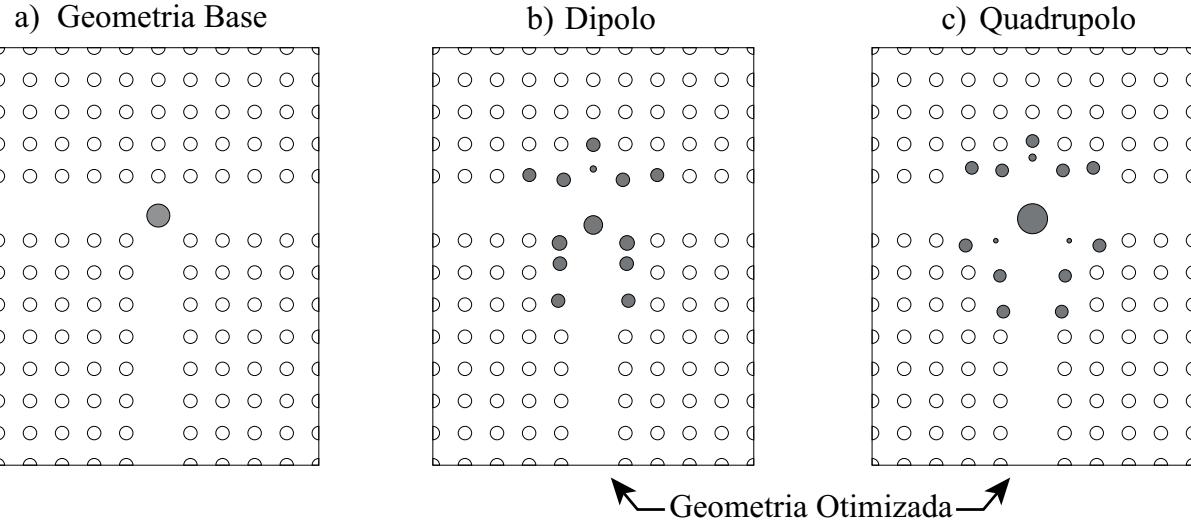


Figura 29 – Geometria dos circuladores após a otimização. a) Geometria base. b) Geometria final do circulador dipolo. c) Geometria final do circulador quadrupolo.

Fonte: do Autor.

A Tabela 2 mostra que para o circulador dipolo foram necessários 641 *Loops* para o algoritmo chegar na resposta em frequência mostrada na Fig. 27. Para o circulador quadrupolo, foram 1487 *Loops* até o algoritmo atingir a resposta em frequência mostrada na Fig. 28.

Tabela 2 – Comparaçāo do desempenho de otimizaçāo dos divisores.

Dispositivo	Loop	MSE
Circulador Dipolo	641	0,0059
Circulador Quadrupolo	1487	0,0083

Fonte: do Autor.

A Fig. 30 mostra a evolução da função custo à medida que o processo exemplificado na Fig. 21 é executado, de forma que o banco de dados do circulador dipolo incrementa até atingir 641 *Loops*. Nesse estágio, a resposta em frequência do dispositivo foi aproximada do espectro desejado, conforme o algoritmo mostrado na Fig. 21.

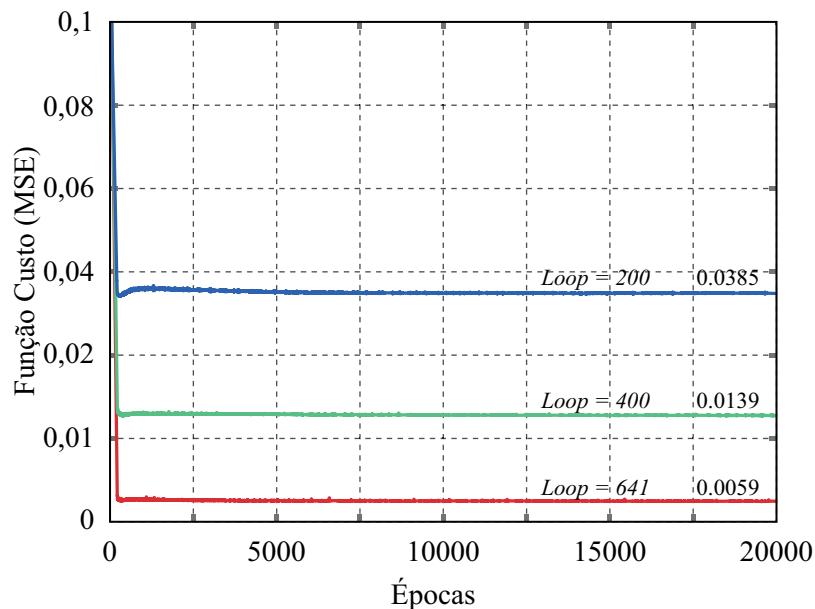


Figura 30 – Evolução da função custo para o circulador dipolo.

Fonte: do Autor.

O mesmo comportamento foi observado no gráfico da avaliação da função custo para o circulador de ressonância quadrupolo, como mostra a Fig. 31, à medida que o banco de dados do referido circulador incrementa até atingir 1487 *Loops*.

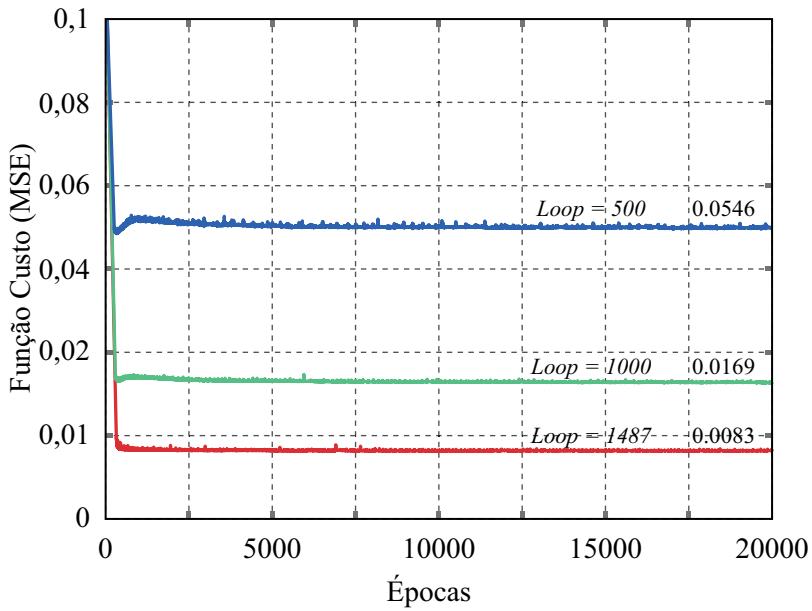


Figura 31 – Evolução da função custo para o circulador quadrupolo.

Fonte: do Autor.

A avaliação da função custo mostrada nas Figs. 30 e 31, são referentes ao desempenho da rede neural em relação ao banco de dados de teste, particionado em 10% (ver Seção 6.2 do Apêndice 6). Nesse sentido, quanto mais informações (dados) a rede neural tem sobre o ambiente (dispositivos estudados), maior é a sua acurácia (portanto, menor o erro) em predizer a geometria dos dispositivos.

4.1.1 Fatores de Qualidade

Cada curva S_{ij} da resposta em frequência foi avaliada através de métricas de desempenho. Desta maneira, foi possível avaliar o erro associado (MSE) entre os valores de desempenho considerados ótimos (valor ideal) e os valores obtidos no processo de otimização por aprendizagem profunda (valor otimizado).

A Tabela 3 mostra os fatores de qualidade para o circulador de ressonância dipolo.

Tabela 3 – Fatores de qualidade avaliados para o circulador de ressonância dipolo.

Fatores	Valor Ideal	Valor Otimizado	Erro Associado (MSE)
ΔF_{11}	94,5 GHz	94,5 GHz	0
ΔF_{21}	94,5 GHz	94,5 GHz	0
ΔF_{31}	94,5 GHz	94,5 GHz	0
ΔF_{12}	94,5 GHz	94,5 GHz	0
ΔF_{22}	94,5 GHz	94,5 GHz	0
ΔF_{32}	94,5 GHz	94,3 GHz	0,04
ΔF_{13}	94,5 GHz	94,4 GHz	0,01
ΔF_{23}	94,5 GHz	94,5 GHz	0
ΔF_{33}	94,5 GHz	94,5 GHz	0
ΔT_{13}	-1 dB	-1,7 dB	0,49
ΔT_{11}	-1 dB	-1,7 dB	0,49
ΔT_{12}	-1 dB	-1,7 dB	0,49
ΔT_{21}	≤ -20 dB	-24,9 dB	0
ΔT_{22}	≤ -20 dB	-19,98 dB	0,04
ΔT_{23}	≤ -20 dB	-24,85 dB	0
ΔT_{21}	≤ -20 dB	-20,1 dB	0
ΔT_{23}	≤ -20 dB	-20,1 dB	0
BW	--	1,5 GHz	--

Fonte: do Autor.

Assim, ΔF_{ij} mostra a ressonância de cada curva em relação à frequência central de operação ($F_c = 94,5$ GHz). A maior parte das curvas foram otimizadas (erro igual a zero), com exceção das curvas ΔF_{32} e ΔF_{13} , com erros de 0,04 e 0,01, respectivamente. As curvas ΔT_{1j} têm por objetivo de otimização serem maximizadas a -1 dB, e em todos os casos houve um erro de 0,49 em relação ao valor obtido de -1,7 dB. Para as curvas ΔT_{2ij} , apenas ΔT_{21} obteve erro diferente de zero (0,04), enquanto que as demais foram otimizadas para um valor abaixo de -20 dB.

Para o circulador de ressonância quadrupolo, os fatores de qualidade estão mostrados na Tabela 4.

Tabela 4 – Fatores de qualidade avaliados para o circulador de ressonância quadrupolo.

Fatores	Valor Ideal	Valor Otimizado	Erro Associado (MSE)
ΔF_{11}	94,5 GHz	94,5 GHz	0
ΔF_{21}	94,5 GHz	94,4 GHz	0,01
ΔF_{31}	94,5 GHz	94,5 GHz	0
ΔF_{12}	94,5 GHz	94,5 GHz	0
ΔF_{22}	94,5 GHz	94,5 GHz	0
ΔF_{32}	94,5 GHz	94,4 GHz	0,01
ΔF_{13}	94,5 GHz	94,4 GHz	0,01
ΔF_{23}	94,5 GHz	94,4 GHz	0,01
ΔF_{33}	94,5 GHz	94,5 GHz	0
ΔT_{121}	-1 dB	-1,76 dB	0,57
ΔT_{132}	-1 dB	-1,72 dB	0,51
ΔT_{113}	-1 dB	-1,76 dB	0,57
ΔT_{211}	\leq -20 dB	-21,87 dB	0
ΔT_{231}	\leq -20 dB	-23,09 dB	0
ΔT_{212}	\leq -20 dB	-23,09 dB	0
ΔT_{222}	\leq -20 dB	-20,97 dB	0
ΔT_{223}	\leq -20 dB	-20,99 dB	0
ΔT_{233}	\leq -20 dB	-21,02 dB	0
BW	--	0,9 GHz	--

Fonte: do Autor.

A otimização das curvas ΔF_{ij} para a frequência central (94,5 GHz) obteve erro de 0,01 para as curvas ΔF_{21} , ΔF_{32} , ΔF_{13} e ΔF_{23} . As curvas ΔT_{1ij} obtiveram erro de 0,57 para as curvas ΔT_{121} e ΔT_{113} e de 0,51, para a curva ΔT_{132} . Todas as curvas ΔT_{2ij} cumpriram o objetivo de otimização para serem minimizadas abaixo de -20 dB.

De forma geral, nas Tabelas 3 e 4, há parâmetros referentes a ΔT_{2ij} que têm erro associado igual a zero, pois cumpriram a condição \leq -20 dB (não importando o quanto menor ΔT_{2ij} for de -20 dB). Neste caso, o erro (MSE) será diferente de zero nas situações em que ΔT_{2ij} for maior que -20 dB.

A largura de banda BW foi calculada considerando os pontos f_1 e f_2 (ver Fig. 23) das curvas mais internas a -15 dB.

5 CONSIDERAÇÕES FINAIS

5.1 Discussão

No âmbito da modelagem inversa, não é sempre garantido que haverá uma solução para o problema, isto é, que tenha de fato uma resposta em frequência com parâmetros de operação desejados. O problema associado é o da *não-unicidade* da resposta eletromagnética de muitos dispositivos. Nesse caso, duas ou mais configurações de geometria podem ocasionar na mesma resposta de campo (ou resposta em frequência). Esse problema, inclusive, foi discutido em [14] e previamente comentado no Capítulo 1 deste trabalho. Deve-se pontuar que, apesar dessa temática não ser abordada na metodologia deste trabalho, é um problema que deve-se considerar para trabalhos futuros.

Outra discussão, é o fato de o banco de dados não ser estático, isto é, há um incremento no número de instâncias à medida que o algoritmo é executado. Por este motivo foi feito o procedimento de gerar um *banco de dados inicial* para posteriormente escolher a arquitetura de rede e, finalmente, continuar com o trabalho.

Durante o Capítulo 3, foi mencionado que a resposta em frequência foi discretizada em 51 pontos. Essa escolha se deu por conta da quantidade de informações que serão repassadas à rede. Quanto mais pontos as curvas forem discretizadas, maior será a dimensão da camada de entrada, mais dados serão processados, o que irá requerer mais tempo. Com menos pontos de discretização, certamente será um processamento mais rápido, entretanto, poderá faltar informações valiosas para o aprendizado, tornando o modelo menos preciso. Desta forma, 51 foi uma escolha considerada ótima entre essas questões discutidas.

A escolha de uma camada de *pré-processamento* na arquitetura da rede neural profunda foi feita com base no trabalho discutido em [13]. Quando implementada, foi verificado um melhoramento na precisão da rede neural profunda. A camada de pré-processamento irá tratar cada curva da resposta em frequência de forma independente e paralela (é como se cada curva tivesse a sua própria rede neural), para então serem alimentadas na rede sequencial.

Os resultados de ambos circuladores baseado em cristal fotônico foi alcançado em até 3 semanas após a implementação do método de otimização e modelagem inversa abordado no presente trabalho. Em contrapartida, em métodos convencionais, esse tempo normalmente é de alguns meses, podendo chegar a alguns anos.

5.2 Conclusão

Neste Trabalho de Conclusão de Curso, foi apresentado um procedimento de otimização e modelagem inversa de dispositivos nanofotônicos com o uso de algoritmos em *machine learning*. Os dispositivos nanofotônicos de hoje dependem cada vez mais de nanoestruturas complexas para realizar funcionalidades sofisticadas. À medida que essa complexidade estrutural aumenta, os processos de projeto se tornam mais desafiadores.

Para ambos os dispositivos baseados em Cristal Fotônico, os resultados foram muito promissores. Conforme relatado no presente trabalho, a caracterização da resposta em frequência de ambos dispositivos ficaram próximas da resposta em frequência desejada, respeitando características de projeto cruciais, como o alinhamento das curvas dos parâmetros-S em ressonância na frequência central de operação dos dispositivos. Para os dispositivos divisores de potência baseados em grafeno (ver Apêndice 6), os resultados ainda não foram promissores e, certamente, requerem uma nova abordagem metodológica (como as demais abordagens de alto DOF discutidas na Seção 1.1 do Capítulo 1). Também é importante considerar que existe a possibilidade, seja pela própria característica eletromagnética do dispositivo, de não existir uma resposta em frequência com características ótimas de operação e, portanto, não existir uma convergência para a modelagem inversa da estrutura analisada.

De uma forma geral, o método desenvolvido se apresentou promissor para a caracterização de nanoestruturas, pois as redes neurais profundas conseguem relacionar muito bem problemas multivariáveis e não-lineares, o que na condição da análise humana através de um processo intuitivo e empírico, torna-se uma tarefa bastante demorada e desafiadora.

O presente trabalho teve como principal objetivo demonstrar ao leitor quais passos e análises seguir para a modelagem inversa de estruturas. Ressalta-se, portanto, que essa metodologia pode ser aplicada a qualquer problema que envolva o *design* de estruturas, independentemente da escala do problema. O uso da API *Comsol LiveLink For Matlab* também é uma ferramenta de projeto crucial, pois automatiza e integra vários processos.

Nota-se que o poder da Inteligência Artificial, em geral, e as abordagens de Aprendizagem Profunda, em particular, têm implicações de grande impacto no campo da nanotecnologia e nanofotônica, como demonstrado no presente trabalho de *Otimização de Dispositivos Baseados em Cristais Fotônicos Usando Métodos em Machine Learning*, e brevemente discutido em trabalhos relacionados na Introdução deste documento. Esse avanço não se restringe somente à tecnologia, mas atinge também as empresas, os setores públicos, o mercado financeiro, etc. Esse novo olhar para a análise dos dados permite tornar os sistemas cada vez mais robustos e eficientes.

5.3 Sugestões Para Trabalhos Futuros

Para trabalhos futuros, são sugeríveis os estudos:

- Desenvolver uma abordagem de modelagem inversa em torno de Redes Neurais Adversárias Generativas (*Generative Adversarial Network (GAN)*).
- Averiguar o uso de Redes Neurais Convolucionais (*Convolutional Neural Networks (CNN)*).

5.4 Trabalhos Desenvolvidos

O presente Trabalho de Conclusão de Curso teve contribuição aos seguintes trabalhos:

- V. Dmitriev, L. Martins, G. Portela and L. H. Assunção, "*Quadrupole resonator mode versus dipole one in photonic crystal ferrite circulators*", Photonics and Nanostructures - Fundamentals and Applications, (2021). [17]
- V. Dmitriev, G. Portela, F. Nobre, W. Castro and L. H. Assunção, "*Nonreciprocal Dynamically Tunable Power Dividers By Three (1x3) Based on Graphene for Terahertz Region*", Optics Communications, (2021) [18].

Para fins de consulta do leitor, o desenvolvimento de todos os *scripts* podem ser consultado no repositório *GitHub*:

- Dispositivos baseados em cristal fotônico [66].
- Dispositivos baseados em Grafeno [67].

REFERÊNCIAS

- [1] J. D. Joannopoulos, S. G. Johnson, J. N. Winn, and R. D. Meade, *Photonic Crystals: Molding the Flow of Light*. Princeton University Press, 2 ed., 2008.
Citado 2 vezes nas páginas [25](#) e [29](#).
- [2] V. Dmitriev and M. N. Kawakatsu, “Nonreciprocal optical divider based on two-dimensional photonic crystal and magneto-optical cavity,” *Applied Optics*, vol. 51, no. 24, pp. 5917–5920, 2012.
Citado 2 vezes nas páginas [25](#) e [47](#).
- [3] V. Dmitriev and G. Portela, “Optical component: nonreciprocal three-way divider based on magneto-optical resonator,” *Applied Optics*, vol. 52, no. 27, pp. 6657–6662, 2013.
Citado 2 vezes nas páginas [25](#) e [47](#).
- [4] V. Dmitriev and W. Castro, “Dynamically controllable graphene terahertz splitters with nonreciprocal properties,” *Applied Optics*, vol. 58, no. 24, pp. 6513–6518, 2019.
Citado 2 vezes nas páginas [25](#) e [47](#).
- [5] O. C. Zienkiewicz, R. L. Taylor, P. Nithiarasu, and J. Zhu, *The finite element method*, vol. 3. McGraw-hill London, 1977.
Citado na página [25](#).
- [6] S. Noureen, M. Zubair, M. Ali, and M. Q. Mahmood, “Deep learning based sequence modeling for optical response retrieval of photonic nanostructures,” in *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*, pp. 289–292, IEEE, 2021.
Citado 2 vezes nas páginas [25](#) e [48](#).
- [7] A. Valkanas and D. Giannacopoulos, “A neural network based electromagnetic simulator,” in *2019 22nd International Conference on the Computation of Electromagnetic Fields (COMPUMAG)*, pp. 1–4, IEEE, 2019.
Citado 2 vezes nas páginas [25](#) e [48](#).
- [8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
Citado 2 vezes nas páginas [25](#) e [35](#).

- [9] M. Al-Qizwini, I. Barjasteh, H. Al-Qassab, and H. Radha, “Deep learning algorithm for autonomous driving using googlenet,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 89–96, IEEE, 2017.
- Citado 2 vezes nas páginas [25](#) e [34](#).
- [10] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- Citado na página [25](#).
- [11] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.
- Citado 2 vezes nas páginas [25](#) e [35](#).
- [12] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708, 2014.
- Citado na página [25](#).
- [13] I. Malkiel, A. Nagler, M. Mrejen, U. Arieli, L. Wolf, and H. Suchowski, “Deep learning for design and retrieval of nano-photonic structures,” *arXiv preprint arXiv:1702.07949*, 2017.
- Citado 4 vezes nas páginas [25](#), [26](#), [49](#) e [69](#).
- [14] D. Liu, Y. Tan, E. Khoram, and Z. Yu, “Training deep neural networks for the inverse design of nanophotonic structures,” *Acs Photonics*, vol. 5, no. 4, pp. 1365–1369, 2018.
- Citado 5 vezes nas páginas [25](#), [26](#), [27](#), [49](#) e [69](#).
- [15] K. Kojima, Y. Tang, T. Koike-Akino, Y. Wang, D. Jha, K. Parsons, M. TaherSima, F. Sang, J. Klamkin, and M. Qi, “Inverse design of nanophotonic devices using deep neural networks,” in *Asia Communications and Photonics Conference (ACP)*, p. Su1A.1, Optical Society of America, 2020.
- Citado na página [25](#).
- [16] J. Peurifoy, Y. Shen, L. Jing, Y. Yang, F. Cano-Renteria, B. G. DeLacy, J. D. Joannopoulos, M. Tegmark, and M. Soljačić, “Nanophotonic particle simulation and inverse design using artificial neural networks,” *Science advances*, vol. 4, no. 6, p. eaar4206, 2018.

Citado 2 vezes nas páginas [25](#) e [49](#).

- [17] V. Dmitriev, L. Martins, G. Portela, and L. Assunção, “Quadrupole resonator mode versus dipole one in photonic crystal ferrite circulators,” *Photonics and Nanostructures - Fundamentals and Applications*, p. 100954, 2021.

Citado 3 vezes nas páginas [26](#), [58](#) e [71](#).

- [18] V. Dmitriev, F. Nobre, W. Castro, G. Portela, and A. Luiz, “Nonreciprocal dynamically tunable power dividers by three (1x3) based on graphene for terahertz region,” *Optics Communications*, 2021.

Citado 3 vezes nas páginas [26](#), [71](#) e [81](#).

- [19] M. H. Tahersima, K. Kojima, T. Koike-Akino, D. Jha, B. Wang, C. Lin, and K. Parsons, “Deep neural network inverse design of integrated photonic power splitters,” *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019.

Citado na página [27](#).

- [20] Z. Liu, D. Zhu, L. Raju, and W. Cai, “Tackling photonic inverse design with machine learning,” *Advanced Science*, vol. 8, no. 5, p. 2002923, 2021.

Citado na página [27](#).

- [21] S. So and J. Rho, “Designing nanophotonic structures using conditional deep convolutional generative adversarial networks,” *Nanophotonics*, vol. 8, no. 7, pp. 1255–1261, 2019.

Citado na página [28](#).

- [22] M. Ohtsu, K. Kobayashi, T. Kawazoe, T. Yatsui, and M. Naruse, *Principles of nanophotonics*. CRC Press, 2008.

Citado 2 vezes nas páginas [29](#) e [31](#).

- [23] Y. Shen and P. Prasad, “Nanophotonics: a new multidisciplinary frontier,” *Applied Physics B*, vol. 74, no. 7, pp. 641–645, 2002.

Citado 2 vezes nas páginas [29](#) e [31](#).

- [24] M. Sumetsky, “Nanophotonics of optical fibers,” *Nanophotonics*, vol. 2, 11 2013.

Citado na página [29](#).

- [25] M. Notomi, “Nanophotonics for optical communications,” in *2011 37th European Conference and Exhibition on Optical Communication*, pp. 1–1, 2011.

Citado na página [29](#).

- [26] Y. Liu, N. Pang, Y. Cai, Y. Yang, C. Zeng, and Y. Wang, “Application of nano optics in photographic imagery and medical imaging,” *Journal of Chemistry*, vol. 2021, 2021.
- Citado na página 29.
- [27] A. Polman, M. Knight, E. C. Garnett, B. Ehrler, and W. C. Sinke, “Photovoltaic materials: Present efficiencies and future challenges,” *Science*, vol. 352, no. 6283, p. aad4424, 2016.
- Citado na página 29.
- [28] C. Kittel, P. McEuen, and P. McEuen, *Introduction to solid state physics*, vol. 8. Wiley New York, 1996.
- Citado na página 29.
- [29] Y. B. Band and Y. Avishai, *Quantum mechanics with applications to nanotechnology and information science*. Academic Press, 2013.
- Citado na página 29.
- [30] W. Ghann and J. Uddin, “Terahertz (thz) spectroscopy: A cutting-edge technology,” *Terahertz Spectroscopy-A Cutting Edge Technology*, 2017.
- Citado 2 vezes nas páginas 29 e 30.
- [31] B. S. Williams, “Terahertz quantum-cascade lasers,” *Nature photonics*, vol. 1, no. 9, pp. 517–525, 2007.
- Citado na página 29.
- [32] K. Fukunaga, I. Hosako, I. Duling III, and M. Picollo, “Terahertz imaging systems: a non-invasive technique for the analysis of paintings,” in *O3A: Optics for Arts, Architecture, and Archaeology II*, vol. 7391, p. 73910D, International Society for Optics and Photonics, 2009.
- Citado na página 30.
- [33] X.-C. Zhang and J. Xu, *Introduction to THz wave photonics*, vol. 29. Springer, 2010.
- Citado na página 30.
- [34] Y. Lu and X. Ning, “A vision of 6g – 5g’s successor,” *Journal of Management Analytics*, vol. 7, no. 3, pp. 301–320, 2020.
- Citado na página 30.
- [35] K. Fukunaga, Y. Ogawa, S. Hayashi, and I. Hosako, “Terahertz spectroscopy for art conservation,” *IEICE Electronics Express*, vol. 4, no. 8, pp. 258–263, 2007.
- Citado na página 30.

- [36] M. Tsurkan and O. Smolyanskaya, “Impact of terahertz radiation on cells,” in *2013 Asia-Pacific Microwave Conference Proceedings (APMC)*, pp. 630–632, IEEE, 2013.
Citado na página 30.
- [37] M. Wang and E.-H. Yang, “Thz application of 2d materials - graphene and beyond,” *Nano-Structures and Nano-Objects*, vol. 15, pp. 107–113, 2018.
Citado na página 31.
- [38] M. S. Vitiello, “Nanodevices at terahertz frequency based on 2d materials,” *Journal of Physics: Materials*, vol. 3, no. 1, p. 014008, 2019.
Citado 2 vezes nas páginas 31 e 32.
- [39] Physicsworld and I. Dumé, “Shaped light waves penetrate further into photonic crystals.” <https://physicsworld.com/a/shaped-light-waves-penetrate-further-into-photonic-crystals/>, 2021. [Acessado em 4/09/2021].
Citado na página 31.
- [40] Nature and E. Gibney, “Surprise graphene discovery could unlock secrets of superconductivity.” <https://www.nature.com/articles/d41586-018-02773-w>, 2018. [Acessado em 4/09/2021].
Citado na página 31.
- [41] E. Yablonovitch, “Photonic band-gap structures,” *JOSA B*, vol. 10, no. 2, pp. 283–295, 1993.
Citado na página 31.
- [42] E. Yablonovitch, “Inhibited spontaneous emission in solid-state physics and electronics,” *Physical review letters*, vol. 58, no. 20, p. 2059, 1987.
Citado na página 31.
- [43] S. John, “Strong localization of photons in certain disordered dielectric superlattices,” *Physical review letters*, vol. 58, no. 23, p. 2486, 1987.
Citado na página 31.
- [44] P. J. Pupalaikis, *S-Parameters for Signal Integrity*. Cambridge University Press, 2020.
Citado na página 34.
- [45] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
Citado na página 34.

- [46] D. Patil and H. Mason, *Data Driven*. "O'Reilly Media, Inc.", 2015.
Citado na página 35.
- [47] C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Springer, 2018.
Citado 5 vezes nas páginas 35, 37, 38, 39 e 40.
- [48] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
Citado na página 35.
- [49] D. Castelvecchi, "Deep learning boosts google translate tool," *Nature News*, 2016.
Citado na página 35.
- [50] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
Citado 2 vezes nas páginas 36 e 37.
- [51] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.
Citado na página 37.
- [52] P. Dayan and L. F. Abbott, *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Computational Neuroscience Series, 2001.
Citado na página 37.
- [53] S. R. y Cajal, *Comparative study of the sensory areas of the human cortex*. Clark University, 1899.
Citado na página 37.
- [54] S. Haykin, *Redes neurais: princípios e prática*. Bookman Editora, 2007.
Citado 7 vezes nas páginas 37, 38, 40, 42, 43, 44 e 46.
- [55] A. F. Agarap, "Deep learning using rectified linear units (relu)," *ArXiv*, vol. abs/1803.08375, 2018.
Citado na página 39.
- [56] G. Lin and W. Shen, "Research on convolutional neural network based on improved relu piecewise activation function," *Procedia computer science*, vol. 131, pp. 977–984, 2018.
Citado na página 39.

- [57] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Always learning, Pearson, 4 ed., 2020.
- Citado 2 vezes nas páginas 41 e 42.
- [58] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proceedings of COMPSTAT’2010*, pp. 177–186, Springer, 2010.
- Citado 2 vezes nas páginas 43 e 52.
- [59] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2 ed., 2019.
- Citado 4 vezes nas páginas 43, 46, 52 e 53.
- [60] V. Dmitriev, G. Portela, and R. Batista, “Magneto-optical resonator switches in two-dimensional photonic crystals: geometry, symmetry, scattering matrices, and two examples,” *Applied Optics*, vol. 53, no. 20, pp. 4460–4467, 2014.
- Citado na página 47.
- [61] M. E. Hines, “Reciprocal and nonreciprocal modes of propagation in ferrite stripline and microstrip devices,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 19, no. 5, pp. 442–451, 1971.
- Citado na página 47.
- [62] COMSOL, “Interface matlab with comsol multiphysics via livelink for matlab.” <https://www.comsol.com/livelink-for-matlab>.
- Citado na página 50.
- [63] D. D. Vvedensky and T. S. Evans, *Symmetry, Groups, and Representations in Physics*. World Scientific, 2009.
- Citado na página 50.
- [64] M. Abadi, A. Agarwal, and et al, “TensorFlow: Large-scale machine learning on heterogeneous systems.” <https://www.tensorflow.org/>.
- Citado na página 53.
- [65] G. Van Rossum and F. L. Drake Jr, *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- Citado na página 53.
- [66] L. H. P. Assunção, “Deep learning in photonic crystal optimization.” <https://github.com/luizheinrich/DeepLearning-PhC>, 2021.
- Citado na página 71.

- [67] L. H. P. Assunção, “Deep learning in graphene optimization.” <https://github.com/luiizheinrich/DeepLearning-Graphene>, 2021.

Citado na página 71.

- [68] F. Nobre, *Divisores de Potência Por Três (1x3) Não-Recíprocos na Faixa de Terahertz Baseado em Grafeno*. PhD thesis, Universidade Federal do Pará, 2021.

Citado 3 vezes nas páginas 81, 85 e 86.

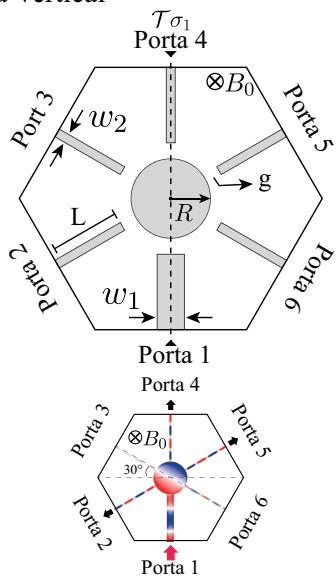
6 APÊNDICES

Além da aplicação nos dispositivos baseados em cristais fotônicos, também foi realizado o procedimento de otimização em dois divisores de potência baseados em grafeno. Na Seção 6.1, é mostrada a metodologia e na Seção 6.1.3, são mostrados os resultados.

6.1 APÊNDICE A – APLICAÇÃO EM DIVISOR DE POTÊNCIA

Os divisores de potência por três (1x3) baseado em grafeno discutidos em [18] foram submetidos ao procedimento deste trabalho. Como mostrado na Fig. 32, o divisor $\mathcal{T}\sigma_1$ tem simetria vertical e o divisor $\mathcal{T}\sigma_2$, simetria horizontal.

a) Simetria Vertical



b) Simetria Horizontal

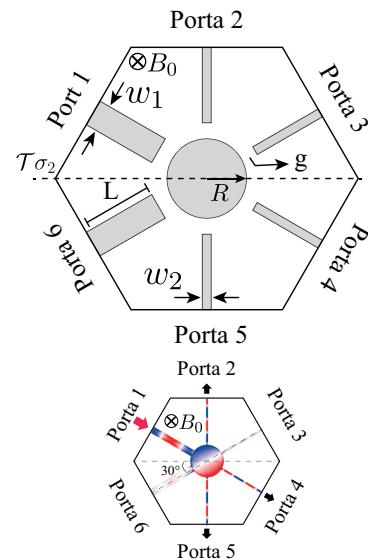
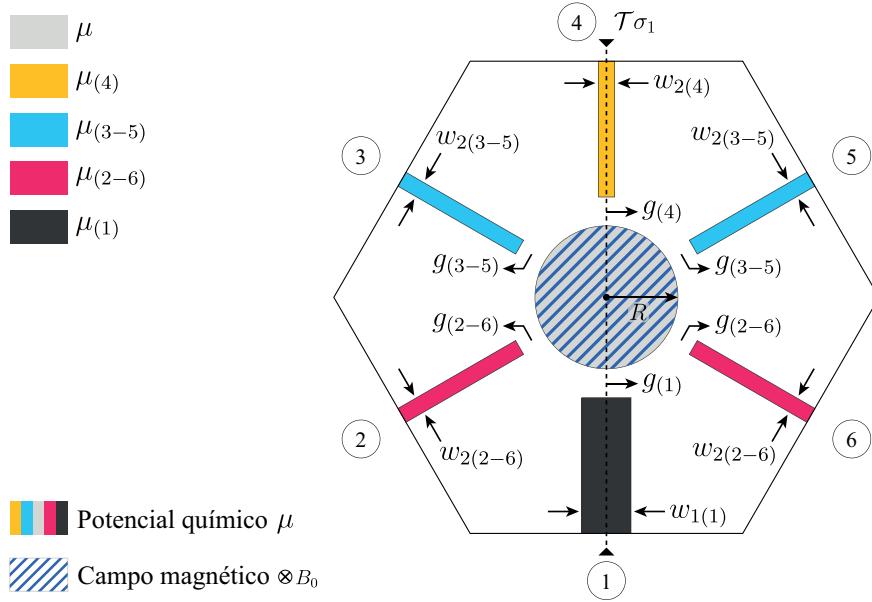


Figura 32 – Divisores de potência suas respectivas distribuições do campo eletromagnético para excitação na porta 1. a) Divisor $\mathcal{T}\sigma_1$. b) Divisor $\mathcal{T}\sigma_2$.

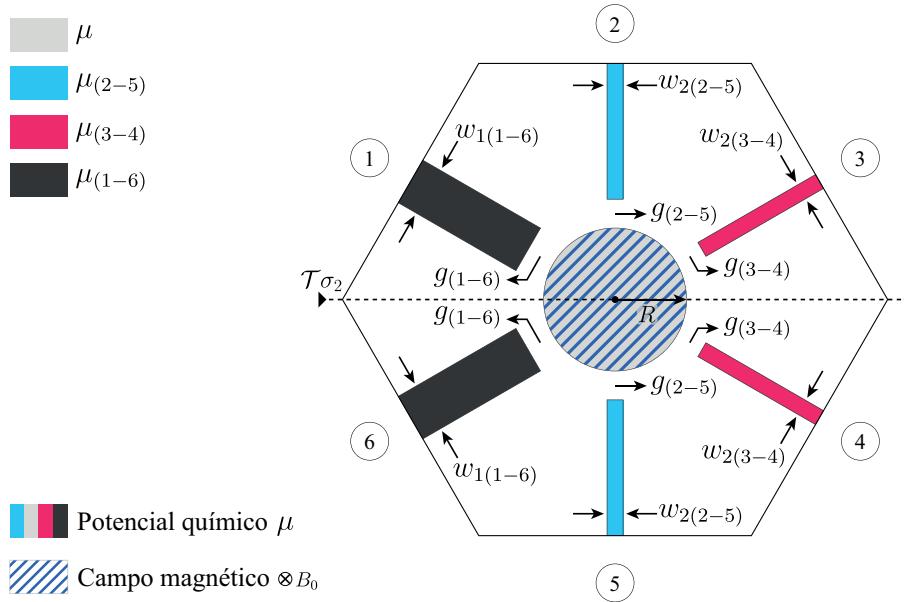
Fonte: Francisco Nobre (2021) (adaptado pelo Autor) [68].

A geometria de ambos foi parametrizada de modo a respeitar a simetria particular de cada dispositivo. Algumas dessas variáveis são w_1 e w_2 (espessura dos guias de onda), R (raio do ressonador), $\otimes B_0$ (campo magnético) e g (gap dos guias de onda).

Figura 33 – Variáveis geométricas do divisor vertical $T\sigma_1$.

Fonte: do Autor.

No total, tanto para $T\sigma_1$ (ver Fig. 33) quanto para $T\sigma_2$ (ver Fig. 34), foram avaliados 19 parâmetros geométricos. Assim, o tensor $[Y]$ terá dimensão $i \times 19$ para ambos dispositivos.

Figura 34 – Variáveis geométricas do divisor horizontal $T\sigma_2$.

Fonte: do Autor.

A resposta em frequência é avaliada considerando o desempenho do sinal da *porta 1* para todas as outras (ver *distribuição do campo eletromagnético* na Fig. 32). Portanto,

6 curvas são avaliadas: S_{11} , S_{21} , S_{31} , S_{41} , S_{51} e S_{61} . Para o divisor $\mathcal{T}\sigma_1$, as curvas S_{21} , S_{41} , S_{51} devem ser maximizadas, pois o sinal deve sair da *porta 1* e se dividir para essas portas, enquanto que as curvas S_{11} , S_{31} , S_{61} devem ser minimizadas (já que o sinal não deve ser propagado para elas). O mesmo raciocínio se extende para o $\mathcal{T}\sigma_2$, sendo as curvas S_{21} , S_{41} e S_{51} maximizadas (sinal propagado) e as curvas S_{11} , S_{31} , S_{61} minimizadas (sinal isolado).

Tanto para o divisor $\mathcal{T}\sigma_1$ quanto para o divisor $\mathcal{T}\sigma_2$, cada curva da resposta em frequência está discretizada em 51 pontos, portanto, $51 \times 6 = 306$. Isso indica que os tensores $[X]$ e $[Z]$ terão dimensão $i \times 306$ e 1×306 , respectivamente. Desta forma, a rede neural profunda terá 306 neurônios na camada de entrada e 19 neurônios na camada de saída para os dois divisores $\mathcal{T}\sigma_1$ e $\mathcal{T}\sigma_2$.

6.1.1 Arquitetura de Rede

Definido a quantidade de neurônios nas camadas de entrada e saída da rede neural profunda, o próximo passo foi definir a arquitetura de rede. A escolha igual das mesmas quantidades de variáveis de geometria para ambos divisores foi intencional, pois facilita o uso de uma arquitetura para trabalhar ambos dispositivos. Foram desenvolvidas 6 arquiteturas de redes com diversas configurações de neurônios e camadas.

- Rede 1: $306 - 150 - 19$
- Rede 2: $306 - 200 - 100 - 19$
- Rede 3: $306 - 150 - 100 - 50 - 19$
- Rede 4: $(51) \parallel \times 6 \rightarrow 306 - 150 - 100 - 50 - 19$
- Rede 5: $(51 - 51 - 51) \parallel \times 6 \rightarrow 306 - 150 - 100 - 50 - 19$
- Rede 6: $(51 - 51 - 51 - 51 - 51 - 51) \parallel \times 6 \rightarrow 306 - 150 - 100 - 50 - 19$

Após a geração do banco de dados inicial i_0 , foi realizado testes de performance das 6 arquiteturas de redes. Primeiramente, foi avaliado o desempenho das três primeiras redes. Foi observado que a *rede 3* obteve o melhor desempenho (isto é, menor erro). Posteriormente, foram geradas mais três arquiteturas de redes (*rede 4*, *rede 5* e *rede 6*), as quais são variações da *rede 3* contendo seis redes de pré-processamento paralelo, variando em 1, 3 e 6 camadas, as quais são conectadas à rede sequencial (*rede 3*) através de uma camada de concatenação¹ (essa arquitetura foi previamente discutida na Fig. 20).

¹ O símbolo \rightarrow é usado para indicar a concatenação.

Tabela 5 – Performance de cada arquitetura de rede em relação ao banco de dados inicial i_0 de cada divisor.

Rede	Divisor $\mathcal{T}\sigma_1$ (MSE)	Divisor $\mathcal{T}\sigma_2$ (MSE)
1	$3,5325e^{-2}$	$5,4233e^{-2}$
2	$3,1709e^{-2}$	$4,8093e^{-2}$
3	$2,6919e^{-2}$	$3,7429e^{-2}$
4	$2,6228e^{-2}$	$4,1021e^{-2}$
5	$2,3523e^{-2}$	$3,6947e^{-2}$
6	$2,6551e^{-2}$	$4,0931e^{-2}$

Fonte: do Autor.

Como pode ser conferido na Tabela 5, arquitetura de rede 5 foi escolhida para ambos divisores, pois obteve o menor erro (MSE) quando treinada com o banco de dados inicial i_0 .

6.1.2 Operação Ideal

A Fig. 35 mostra a resposta em frequência ideal designada para ambos divisores, a qual será o objetivo da rede neural. Todas as curvas estão em ressonância na frequência central de operação dos dispositivos.

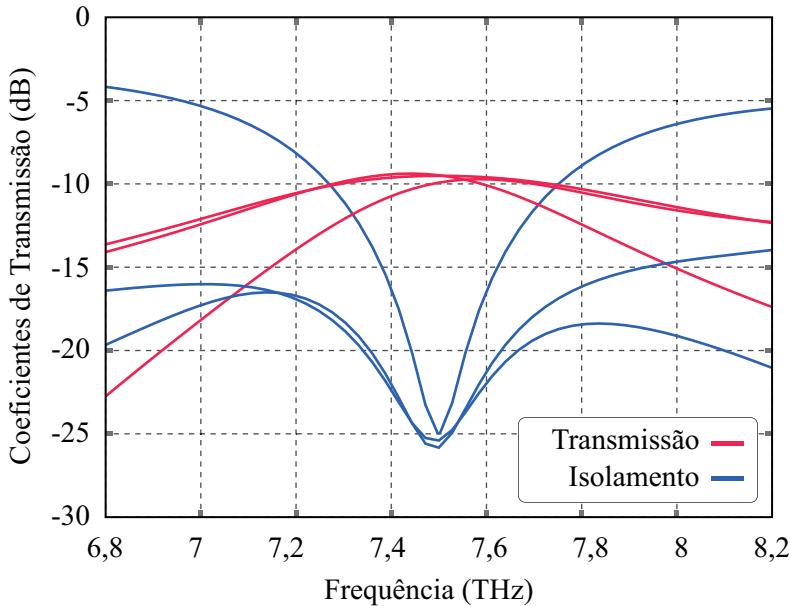


Figura 35 – Resposta em frequência desejada.

Fonte: do Autor.

6.1.3 Resultados

O objetivo da rede neural profunda foi otimizar e realizar a otimização da geometria do divisor vertical a partir da *resposta em frequência desejada* mostrada na Fig. 35. A

Fig. 36 mostra a resposta em frequência do divisor $\mathcal{T}\sigma_1$ após esse processo de otimização.

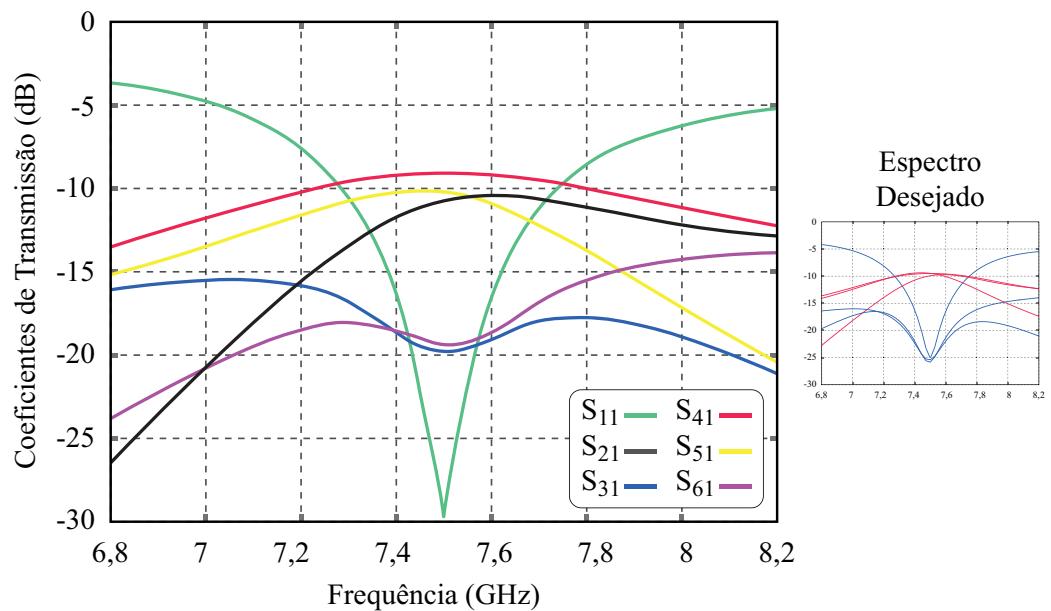


Figura 36 – Resultado da resposta em frequência do divisor vertical $\mathcal{T}\sigma_1$.

Fonte: Francisco Nobre (2021) (adaptado pelo Autor) [68].

Nota-se que a ressonância ocorre na frequência central do divisor $\mathcal{T}\sigma_1$, entretanto, um valor ótimo das curvas de isolamento seria abaixo dos -20dB , requisito que ainda não foi respeitado pelas curvas S_{31} e S_{61} .

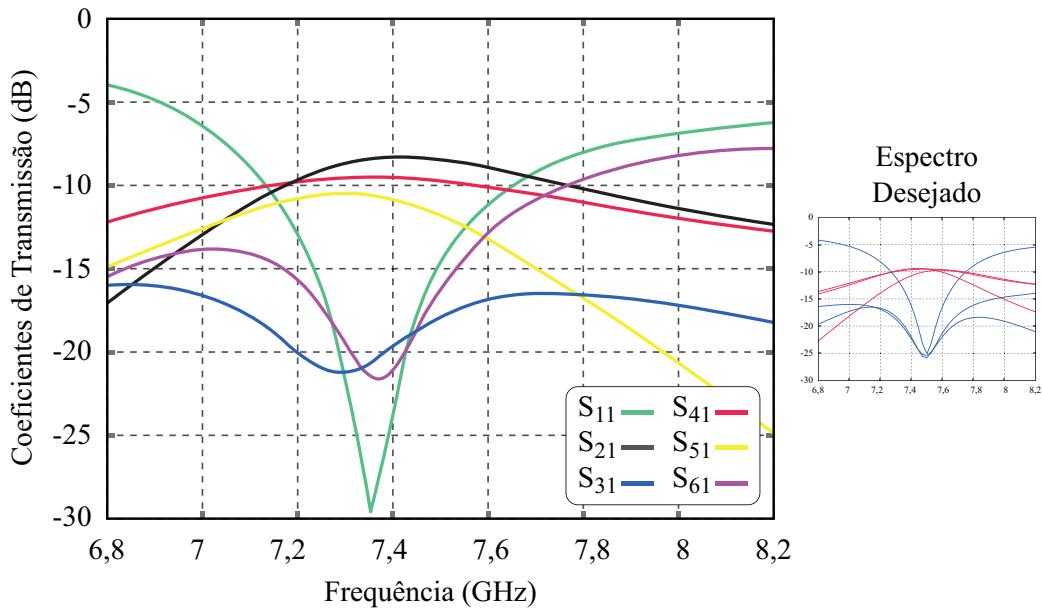
Os parâmetros encontrados pela rede neural (*Valor Otimizado*) para o divisor vertical $\mathcal{T}\sigma_1$ podem ser comparados com a situação do dispositivo antes da otimização (*Valor Original*), como mostrado na Tabela 6.

Tabela 6 – Comparação dos parâmetros do divisor vertical.

Parâmetros	Valor Original	Valor Otimizado	Descrição
$w_{1(1)}$	120nm	117,1798nm	Espessura do guia 1
$w_{2(2-6)}$	28nm	28,0957nm	Espessura dos guia 2 e 6
$C_{(2-6)}$	0°	$+1,4927^\circ$	Contorno dos guias 2 e 6
$w_{2(3-5)}$	28nm	32,8879nm	Espessura dos guia 3 e 5
$C_{(3-5)}$	0°	$+4,8879^\circ$	Contorno dos guias 3 e 5
$w_{2(4)}$	28nm	28,5929nm	Espessura do guia 4
μ	0,15eV	0,1571eV	Potencial químico do ressonador
$\mu_{(1)}$	0,15eV	0,1657eV	Potencial químico do guia 1
$\mu_{(2-6)}$	0,15eV	0,1397eV	Potencial químico dos guias 2 e 6
$\mu_{(3-5)}$	0,15eV	0,1631eV	Potencial químico dos guias 3 e 5
$\mu_{(4)}$	0,15eV	0,1585eV	Potencial químico do guia 4
B_0	0,29T	0,2079T	Campo magnético
R	320nm	328,8418nm	Raio do ressonador
$g_{(1)}$	2,5nm	2,3263nm	Gap do guia 1
$g_{(2-6)}$	2,5nm	1,7391nm	Gap dos guias 2 e 6
$g_{(3-5)}$	2,5nm	3,6179nm	Gap dos guias 3 e 5
$g_{(4)}$	2,5nm	4,4295nm	Gap do guia 4

Fonte: Francisco Nobre (2021) [68].

A resposta em frequência do divisor de simetria horizontal $\mathcal{T}\sigma_2$ após o procedimento de otimização está ilustrada na Fig. 37.

Figura 37 – Resultado da resposta em frequência do divisor horizontal $\mathcal{T}\sigma_2$.

Fonte: Francisco Nobre (2021) (adaptado pelo Autor) [68].

Nota-se que o procedimento ainda não foi satisfatório, pois as curvas não estão em

ressonância na frequência central de operação do divisor $\mathcal{T}\sigma_2$. No total, foram realizados 215 *Loops* para o divisor $\mathcal{T}\sigma_1$ e 634 *Loops* para o divisor $\mathcal{T}\sigma_2$, como mostrado na Tabela 7.

Tabela 7 – Comparaçāo do desempenho de otimizaçāo dos divisores.

Dispositivo	Loop	MSE	Descrição
$\mathcal{T}\sigma_1$	215	0,007188	Divisor Vertical
$\mathcal{T}\sigma_2$	634	0,038134	Divisor Horizontal

Fonte: do Autor.

6.2 APÊNDICE B – CRISTAL FOTÔNICO: PARÂMETROS GE-RAIS

Esta Seção do Apêndice detalha a geometria dos dispositivos baseados em cristal fotônico e as características de projeto da rede neural profunda implementada.

6.2.1 Características da Geometria

A estrutura geométrica dos dispositivos baseados em cristais fotônicos está apresentada na Fig. 38. Ela é composta por cilindros em uma estrutura periódica, cuja constante de rede a é de 0,12 mm. Cada cilindro tem o raio de $0,2a$ mm e posição (centro do cilindro em coordenadas cartesianas (x, y)) em valores discretos da constante de rede.

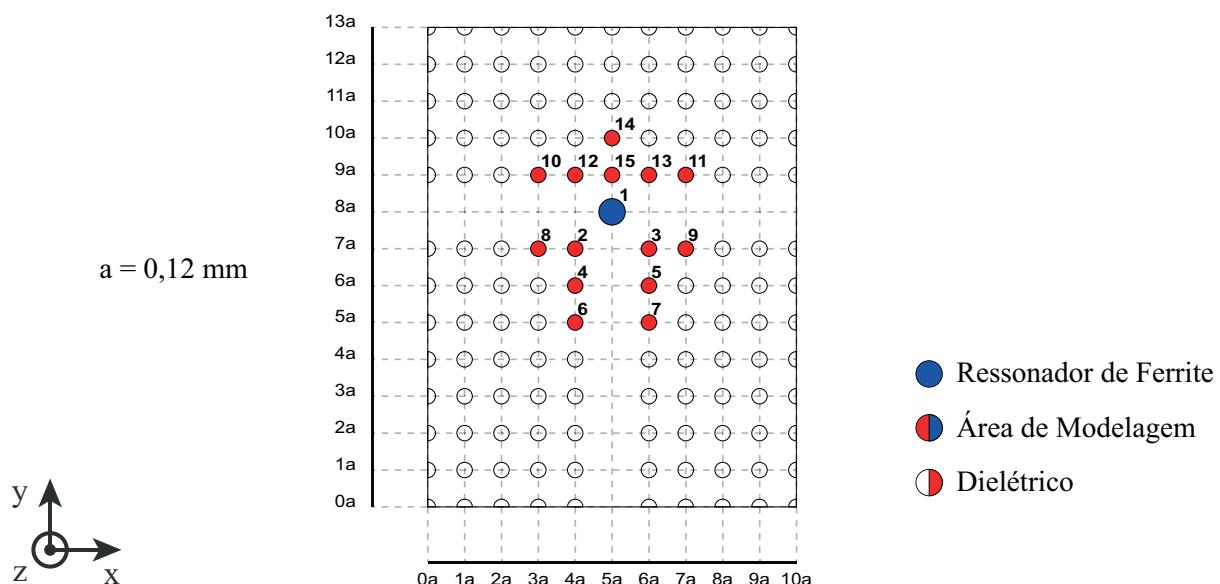


Figura 38 – Parametrização da geometria base do cristal fotônico.

Fonte: do Autor.

A ferrite, localizada no centro da *junção-T*, tem raio específico para cada dispositivo. Para o dispositivo de ressonância dipolo, o raio da ferrite é de $0,2a$ mm, enquanto que para o dispositivo de ressonância quadrupolo, o raio da ferrite é de $0,4a$ mm. A ferrite posta na cavidade ressonante é magnetizada por um campo magnético externo e irá apresentar *dois* ou *quatro* polos, a depender do raio que a ferrite apresentar.

A Tabela 8 mostra os valores de geometria do dispositivo dipolo em milímetros. O *valor original* refere-se à geometria base e o *valor otimizado* à geometria otimizada, como mostrado na Fig. 29.a) e 29.b), respectivamente.

Tabela 8 – Parâmetros otimizados do circulador dipolo.

Cilindro	Valor Original		Valor Otimizado	
	Centro (x, y)	Raio	Centro (x, y)	Raio
1	(5a, 8a)	0,2a	(5a, 7,438a)	0,2854a
2	(4a, 7a)	0,2a	(4,9399a, 6,9244a)	0,2228a
3	(6a, 7a)	0,2a	(6,061a, 6,9244a)	0,2228a
4	(4a, 6a)	0,2a	(4,9518a, 6,1150a)	0,2060a
5	(6a, 6a)	0,2a	(6,0482a, 6,1150a)	0,2060a
6	(4a, 5a)	0,2a	(4,996a, 5,1150a)	0,2a
7	(6a, 5a)	0,2a	(6,0996a, 5,1150a)	0,2a
8	(3a, 7a)	0,2a	(3a, 7a)	0,2a
9	(7a, 7a)	0,2a	(7a, 7a)	0,2a
10	(3a, 9a)	0,2a	(3a, 9,0384a)	0,2023a
11	(7a, 9a)	0,2a	(7a, 9,0384a)	0,2023a
12	(4a, 9a)	0,2a	(4,0786a, 8,8963a)	0,2012a
13	(6a, 9a)	0,2a	(5,9214a, 8,8963a)	0,2012a
14	(5a, 10a)	0,2a	(5a, 9,2305a)	0,0953a
15	(5a, 9a)	0,2a	(5a, 9,9823a)	0,2a

Fonte: do Autor.

Na Tabela 9, o *valor original* também refere-se à geometria base e o *valor otimizado* à geometria otimizada para o circulador quadrupolo, como mostrado na Fig. 29.a) e 29.c), respectivamente.

Tabela 9 – Parâmetros otimizados do circulador quadrupolo.

Cilindro	Valor Original		Valor Otimizado	
	Centro (x, y)	Raio	Centro (x, y)	Raio
1	(5a, 8a)	0,4a	(5a, 7,68491a)	0,461478a
2	(4a, 7a)	0,2a	(3,849055a, 6,9923255a)	0,0637046a
3	(6a, 7a)	0,2a	(6,150944a, 6,9923255a)	0,0637046a
4	(4a, 6a)	0,2a	(3,983052a, 5,8950494a)	0,20221a
5	(6a, 6a)	0,2a	(6,016947a, 5,8950494a)	0,20221a
6	(4a, 5a)	0,2a	(4,085462a, 4,780541082a)	0,39467277a
7	(6a, 5a)	0,2a	(5,914537a, 4,780541082a)	0,39467277a
8	(3a, 7a)	0,2a	(2,908861a, 6,84256803a)	0,19825571a
9	(7a, 7a)	0,2a	(7,091138a, 6,84256803a)	0,19825571a
10	(3a, 9a)	0,2a	(3,100644a, 9,26410546a)	0,19975595a
11	(7a, 9a)	0,2a	(6,89935599a, 9,26410546a)	0,19975595a
12	(4a, 9a)	0,2a	(4,05160417a, 9,18938524a)	0,20172808a
13	(6a, 9a)	0,2a	(5,94839582a, 9,18938524a)	0,20172808a
14	(5a, 10a)	0,2a	(5a, 10,10774003a)	0,20051608a
15	(5a, 9a)	0,2a	(5a, 9,58835201a)	0,10690515a

Fonte: do Autor.

6.2.2 Hiperparâmetros da Rede Neural

A rede neural profunda usada para ambos dispositivos baseados em cristais fotônicos tem os seus hiperparâmetros mostrados na Tabela 10.

Tabela 10 – Hiperparâmetros da rede neural.

Hiperparâmetro	Atributo
Otimizador	ADAM; $\eta = 0,001$
Função Custo	MSE
Função de Ativação	LeakyReLU; $\alpha = 0,3$
Batch Size	100
Épocas	20000

Fonte: do Autor.

Assim, η refere-se a taxa de aprendizagem do otimizador ADAM, enquanto que α refere-se ao coeficiente de inclinação negativo da função de ativação LeakyReLU.

A Tabela 11 mostra a divisão percentual do banco de dados em *treino*, *validação* e *teste*.

Tabela 11 – Divisão do banco de dados.

Composição	Atributo
Treino	80%
Validação	10%
Teste	10%

Fonte: do Autor.

6.2.3 Configurações da Máquina

As configurações do computador onde todo o procedimento de otimização foi rodado, estão mostradas na Tabela 12.

Tabela 12 – Configurações do computador.

Característica	Atributo
CPU	Intel Core i5-3210M
GPU	NVIDIA GeForce GT-630M
Memória	8 GB
Armazenamento	SSD 250 GB
Sistema Operacional	Windows 10
COMSOL	Versão 5.5
MATLAB	Versão R2020B
Tempo 1	15 minutos
Tempo 2	20 minutos
Tempo 3	3 semanas - 5 semanas

Fonte: do Autor.

Desta maneira, o *Tempo 1* refere-se ao tempo demandado para o computador especificado rodar uma simulação do COMSOL, enquanto que o *Tempo 2* refere-se ao tempo necessário para treinar a rede neural. O *Tempo 3* foi o tempo total gasto para todo o processo de otimização e modelagem inversa apresentado neste trabalho. O dispositivo circulador de ressonância dipolo demandou 3 semanas, enquanto que o dispositivo circulador de ressonância quadrupolo demandou 5 semanas. É válido ressaltar que o *Tempo 3* não ocorreu de forma ininterrupta.