

UNIVERSIDADE FEDERAL DE SÃO CARLOS — *CAMPUS* SOROCABA
CENTRO DE CIÊNCIAS E TECNOLOGIAS PARA A SUSTENTABILIDADE
DEPARTAMENTO DE FÍSICA, QUÍMICA E MATEMÁTICA

Simulação do Módulo de Young de Compósitos Reforçados com Fibra usando Aprendizado de Máquina

Gabriel Victor Ribeiro

Sorocaba

Agosto, 2022

UNIVERSIDADE FEDERAL DE SÃO CARLOS — *CAMPUS* SOROCABA
CENTRO DE CIÊNCIAS E TECNOLOGIAS PARA A SUSTENTABILIDADE
DEPARTAMENTO DE FÍSICA, QUÍMICA E MATEMÁTICA

Simulação do Módulo de Young de Compósitos Reforçados com Fibra usando Aprendizado de Máquina

Gabriel Victor Ribeiro

Trabalho de Conclusão de Curso apresentado ao curso de Licenciatura Plena em Física da Universidade Federal de São Carlos, *campus* Sorocaba, para obtenção do título de Licenciado em Física.

Orientador: Prof. Dr. Renato Fernandes Cantão

Sorocaba

Agosto, 2022

Ribeiro, Gabriel Victor

Simulação do módulo de Young de compósitos reforçados com fibra usando aprendizado de máquina / Gabriel Victor Ribeiro -- 2022.
80f.

TCC (Graduação) - Universidade Federal de São Carlos, campus Sorocaba, Sorocaba
Orientador (a): Renato Fernandes Cantão
Banca Examinadora: Wallace Gusmão Ferreira, Adriana de Oliveira Delgado Silva
Bibliografia

1. Compósitos reforçados por fibra. 2. Aprendizado de máquina. 3. Redes neurais. I. Ribeiro, Gabriel Victor. II. Título.

Ficha catalográfica desenvolvida pela Secretaria Geral de Informática (SIn)

DADOS FORNECIDOS PELO AUTOR

Bibliotecário responsável: Maria Aparecida de Lourdes Mariano -
CRB/8 6979



UNIVERSIDADE FEDERAL DE SÃO CARLOS - *Campus Sorocaba*
Coordenação do Curso de Licenciatura em Física

Trabalho de Conclusão de Curso

Folha de aprovação

Gabriel Victor Ribeiro

“SIMULAÇÃO DO MÓDULO DE YOUNG DE COMPÓSITOS REFORÇADOS COM
FIBRA USANDO APRENDIZADO DE MÁQUINA”

Documento assinado digitalmente
 RENATO FERNANDES CANTAO
Data: 03/08/2022 16:13:56-0300
Verifique em <https://verificador.iti.br>

Orientador _____
Prof. Dr. Renato Fernandes Cantão

Documento assinado digitalmente
 WALLACE GUSMAO FERREIRA
Data: 03/08/2022 18:35:22-0300
Verifique em <https://verificador.iti.br>

Membro 1 _____
Prof. Dr. Wallace Gusmão Ferreira

Documento assinado digitalmente
 ADRIANA DE OLIVEIRA DELGADO SILVA
Data: 03/08/2022 16:27:28-0300
Verifique em <https://verificador.iti.br>

Membro 2 _____
Prof^a. Dra. Adriana de Oliveira Delgado Silva

Sorocaba, 03 de agosto de 2022.



UNIVERSIDADE FEDERAL DE SÃO CARLOS
CAMPUS SOROCABA
COORDENAÇÃO DO CURSO DE LICENCIATURA EM FÍSICA -
SOROCABA

Rodovia João Leme dos Santos, Km 110 - SP-264
Bairro do Itinga - Sorocaba - São Paulo - Brasil
CEP 18052-780
Telefone: (15) 3229-8859

ATA DA DEFESA PÚBLICA

ATA Nº 05/2022

Aos três dias do mês de agosto de 2022, por videoconferência pela plataforma do Google Meet, realizou-se a defesa pública do trabalho de Conclusão de Curso do estudante **Gabriel Victor Ribeiro** do Curso de Licenciatura em Física – Sorocaba, devidamente matriculado na disciplina de Trabalho de Conclusão de Curso, perante a Banca Examinadora, composta pelos Prof. Dr. Renato Fernandes Cantão, Prof. Dr. Wallace Gusmão Ferreira e Prof^ª. Dr^ª. Adriana de Oliveira Delgado Silva segundo o estabelecido nas Normas para apresentação de Trabalho de Conclusão do Curso.

Após a apresentação e arguições, a Banca deliberou, segundo os critérios estabelecidos nas normas do TCC do curso:

Documento assinado digitalmente
 RENATO FERNANDES CANTAO
Data: 03/08/2022 16:11:08-0300
Verifique em <https://verificador.itl.br>

Nota: **10,0**

Prof. Dr. Renato Fernandes Cantão

Documento assinado digitalmente
 WALLACE GUSMAO FERREIRA
Data: 03/08/2022 18:33:19-0300
Verifique em <https://verificador.itl.br>

Nota: **10,0**

Prof. Dr. Wallace Gusmão Ferreira

Documento assinado digitalmente
 ADRIANA DE OLIVEIRA DELGADO SILVA
Data: 03/08/2022 16:24:01-0300
Verifique em <https://verificador.itl.br>

Nota: **10,0**

e Prof^ª. Dr^ª. Adriana de Oliveira Delgado Silva

Com isso, o Trabalho foi considerado **aprovado** com nota final **10,0**.

Sorocaba, 03 de agosto de 2022.

Dedico este trabalho à minha mãe, Zélia.

Agradecimentos

A minha mãe e minha irmã, pelo apoio, pelo incentivo e pelo amor incondicional, que me permitiu realizar esse sonho.

Aos meus amigos de infância, Bruno e Gustavo, que sempre estiveram comigo desde sempre, obrigado por tudo, vocês sempre fazem eu dar meu melhor.

A todos amigos que fiz na universidade, mas principalmente aos que se tornaram meus irmãos, Alessandro e Vinicius, vocês fizeram esses anos de Universidade valerem a pena.

Aos meus parceiros, Allan, Guilherme e Samuel, por todos os momentos de descontração e apoio, vocês fazem tudo ficar mais leve.

A minha namorada Jéssica, que sem ela seria impossível terminar esse trabalho, obrigado por toda paciência, todo amor, carinho e apoio em todos esses anos.

Ao meu orientador, Prof. Dr. Renato Fernandes Cantão, que foi muito mais do que apenas um professor ou um orientador pra mim. Muito obrigado por todas as oportunidades que me deu, e por todos os papos, espero um dia poder ser tão incrível para um aluno quanto você foi para mim.

Aos demais professores que me ensinaram muito nesses anos de Universidade, seja dentro ou fora de sala de aula.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo apoio financeiro para que eu pudesse realizar minha iniciação científica.

Ao Prof. Dr. Wallace Gusmão Ferreira, pelas dicas e pelo acesso à licença educacional do Altair HyperWorks da UFABC que contribuíram imensamente para que esse trabalho fosse realizado.

Aos técnicos-administrativos e demais membros da comunidade da Universidade Federal de São Carlos por trabalharem diariamente de maneira a permitir que eu concluísse esse curso.

Aos membros da banca, por me darem a honra de ter o trabalho avaliado por professores os quais admiro muito.

*“São as perguntas que não sabemos responder que mais nos ensinam.
Elas nos ensinam a pensar.
Se você dá uma resposta a um homem, tudo o que ele ganha é um fato qualquer.
Mas, se você lhe der uma pergunta, ele procurará suas próprias respostas.”*

(Patrick Rothfuss)

Resumo

O módulo de Young, ou módulo de elasticidade, é uma propriedade que mede a relação de proporcionalidade entre tensões e deformações de um material em regime elástico. Quando trata-se de um material compósito — um material que consiste na combinação de duas diferentes fases: matriz e reforço — o cálculo dessa propriedade nem sempre é simples. Assim, nesse trabalho propõem-se utilizar de aprendizado de máquina, a partir de redes neurais, para estimar o módulo de Young. Com o intuito de obter dados para o treinamento da rede utilizou-se um método bem estabelecido na área, a modelagem multiescala, a partir de simulações no software licenciado *Multiscale Designer*. Como ele foram gerados 152 módulos de Young diferentes, categorizados por porcentagem de fibra, estrutura laminada e tipo de ensaio (compressão ou tração). Para descobrir qual a melhor arquitetura possível para a rede, foram testadas sete diferentes configurações, alterando número de camadas, número de neurônios e função de ativação. Com os testes feitos, definiu-se qual rede apresentava os melhores resultados, e suas previsões foram então comparadas com dados de simulação e experimentais, obtendo erros relativos que não ultrapassam 20 % na maioria dos casos. Os resultados da aplicação de aprendizado de máquina também são mais eficientes computacionalmente, gerando resultados em menos de um segundo para diversas configurações simultaneamente, o que é uma vantagem quando comparado ao software licenciado, que demora na ordem dos minutos para um único material, e quando comparado ao experimental, que exige a manufatura do material.

Palavras-chave: Compósitos reforçados por fibra. Aprendizado de Máquina. Redes neurais.

Abstract

Young's modulus, or modulus of elasticity, is a property that measures the proportionality relationship between stresses and strains of a material in an elastic regime. When dealing with a composite material — a material consisting of the combination of two different phases: matrix and reinforcement — its calculation is not always simple. For this reason, in this paper it is proposed the use of machine learning, specifically neural networks, to estimate Young's modulus. A well established method, the multiscale modeling, was used to generate training data for the neural networks. Simulations using the licensed software *Multiscale Designer*, resulted in 152 different Young's modules, categorized by fiber percentage, laminated structure and test type (compression or tension). To define the best possible architecture for the network, seven different configurations were tested by varying the number of layers, number of neurons and activation function. From the results it was possible to pinpoint the best network, and further compare its predictions with simulated and experimental data, resulting in relative errors that in overall do not exceed 20 %. The results from machine learning are also more computationally efficient, obtaining results in less than a second for several configurations simultaneously, which is an advantage when compared to licensed software, which takes on the order of minutes for a single material, and when compared to the experimental one, which requires manufacturing the material.

Keywords: Fiber-reinforced composites. Machine Learning. Neural Networks.

Lista de Figuras

Figura 1	–	Esquema ilustrativo das diferentes classificações de um material compósito .	17
Figura 2	–	Divisão percentual do uso de compósitos por setor produtivo.	19
Figura 3	–	Presença percentual dos diversos tipos de materiais na estrutura dos aviões Boeing A787. Os compósitos respondem por 50 % do total.	20
Figura 4	–	Níveis de uma análise multiescala de materiais compósitos.	22
Figura 5	–	Esquema ilustrativo do balanço entre eficiência e fidelidade de diferentes métodos da modelagem multiescala.	22
Figura 6	–	Exemplo de diferentes representações de um mesmo conjunto de dados à serem separados em duas categorias por uma linha.	26
Figura 7	–	Exemplo de um modelo de <i>Deep Learning</i> de reconhecimento de imagem. . . .	28
Figura 8	–	Diagrama ilustrativo do funcionamento de um modelo de <i>Deep Learning</i> . A camada de entrada recebe os dados de treinamento x , que serão transformados nas n camadas ocultas, ponderados pelos respectivos pesos $w_i, i = 1, \dots, n$. Na última cada será produzida uma previsão \hat{y} , que será comparada com os valores esperados y usando-se uma função de perda. O resultado dessa comparação é usado pelo otimizador para atualizar os pesos e reiniciar o processo.	32
Figura 9	–	Função da unidade linear retificada, ReLU.	33
Figura 10	–	Ilustração dos elementos típicos de uma rede neural	34
Figura 11	–	Ilustração do que é um <i>underfitting</i> (à esquerda), um bom ajuste (ao centro), e um <i>overfitting</i> (à direita).	38
Figura 12	–	Diferentes técnicas de metamodelagem.	41
Figura 13	–	Duas formas de construção de um compósito: em (a) tem-se uma estrutura laminada, composta de lâminas com fibras em ângulos variados e em (b) tem-se a estrutura em sanduíche.	44
Figura 14	–	Efeito no módulo de Young para cargas em diferentes ângulos para compósitos de dobra cruzada (<i>cross-ply</i>) e quase-isotrópicos.	47
Figura 15	–	Tela inicial do <i>Multiscale Designer</i> , da Altair.	49
Figura 16	–	Tela de definição do modelo de célula unitária do <i>Multiscale Designer</i> , da Altair.	50
Figura 17	–	Malha de um RVE com 40 % de fibras gerado no <i>Multiscale Designer</i> , da Altair.	51
Figura 18	–	Tela de caracterização linear do material do <i>Multiscale Designer</i> , da Altair.	52
Figura 19	–	Representação ilustrativa da configuração de fibras na estrutura laminada.	52
Figura 20	–	Tela de caracterização não linear do material do <i>Multiscale Designer</i> , da Altair.	53
Figura 21	–	Gráfico de correlação nos dados de teste.	58

Figura 22 – Gráfico de correlação para um ensaio de compressão.	60
Figura 23 – Gráfico de correlação para um ensaio de tração.	61
Figura 24 – Gráfico de correlação para um ensaio de compressão após a introdução de um dado categórico que discrimina entre problemas de compressão e de tração nos dados de teste.	64
Figura 25 – Gráfico de correlação para um ensaio de tração após a introdução de um dado categórico que discrimina entre problemas de compressão e de tração nos dados de teste.	65
Figura 26 – Gráfico de correlação para um ensaio de compressão após a introdução de um dado categórico que discrimina entre problemas de compressão e de tração.	66
Figura 27 – Gráfico de correlação para um ensaio de tração após a introdução de um dado categórico que discrimina entre problemas de compressão e de tração.	67
Figura 28 – Erros ao longo das épocas Modelo 1	79
Figura 29 – Erros ao longo das épocas Modelo 2	79
Figura 30 – Erros ao longo das épocas Modelo 3	80
Figura 31 – Erros ao longo das épocas Modelo 4	80
Figura 32 – Erros ao longo das épocas Modelo 5	80
Figura 33 – Erros ao longo das épocas Modelo 6	81
Figura 34 – Erros ao longo das épocas Modelo 7	81
Figura 35 – Erros ao longo das épocas Modelo 4 com dados categóricos	81
Figura 36 – Erros ao longo das épocas Modelo 5 com dados categóricos	82
Figura 37 – Erros ao longo das épocas Modelo 6 com dados categóricos	82

Lista de Tabelas

Tabela 1	– Configurações de matriz e fibra simuladas no <i>Multiscale Designer</i> . A configuração 4 também recebe o nome de <i>quase-isotrópica</i> . As escolhas das estruturas foram baseadas em Wollschlager (2020).	51
Tabela 2	– Topologia das redes neurais testadas para previsão do módulo de Young.	54
Tabela 3	– Otimizador e métricas utilizadas nas redes	55
Tabela 4	– Configurações da máquina, e versões dos softwares utilizados.	55
Tabela 5	– Erros médios absoluto, quadrático e absoluto percentual — respectivamente MAE, MSE e MAPE — e coeficiente de determinação R^2 das redes treinadas.	57
Tabela 6	– Erros percentuais para o ensaio de compressão — Modelo 4	62
Tabela 7	– Erros percentuais para o ensaio de tração — Modelo 4	62
Tabela 8	– Erros percentuais para o ensaio de compressão — Modelo 5	62
Tabela 9	– Erros percentuais para o ensaio de tração — Modelo 5	62
Tabela 10	– Erros percentuais para o ensaio de compressão — Modelo 6	63
Tabela 11	– Erros percentuais para o ensaio de tração — Modelo 6	63
Tabela 12	– Erros relativos médios dos modelos nos dados inéditos em %.	63
Tabela 13	– Erros médios absoluto, quadrático e absoluto percentual — respectivamente MAE, MSE e MAPE — das redes treinadas, agora incluindo um dado categórico para discriminação dos ensaios de compressão e tensão.	63
Tabela 14	– Erros percentuais para o ensaio de compressão — Modelo 4	68
Tabela 15	– Erros percentuais para o ensaio de tração — Modelo 4	68
Tabela 16	– Erros percentuais para o ensaio de compressão — Modelo 5	68
Tabela 17	– Erros percentuais para o ensaio de tração — Modelo 5	68
Tabela 18	– Erros percentuais para o ensaio de compressão — Modelo 6	68
Tabela 19	– Erros percentuais para o ensaio de tração — Modelo 6	69
Tabela 20	– Erros relativos médios dos modelos nos dados inéditos em %.	69
Tabela 21	– Ensaio de tração para comparação do módulo de Young (em GPa) entre o valor previsto pelo modelo 6 e o resultado experimental de Marlett (2011).	69
Tabela 22	– Ensaio de compressão para comparação do módulo de Young (em GPa) entre o valor previsto pelo modelo 6 e o resultado experimental de Marlett (2011).	70

Conteúdo

1	INTRODUÇÃO	15
1.1	Materiais compósitos	15
1.2	Caracterização	17
1.3	Aplicações	18
1.4	Modelagem matemática	20
2	APRENDIZADO DE MÁQUINA	24
2.1	Contexto histórico	24
2.1.1	O nascimento da IA	24
2.1.2	Aprendizado de Máquina	25
2.1.3	Aprendizado Profundo	27
2.1.4	Conectismo	29
2.1.5	<i>Deep Belief Networks</i>	30
2.2	<i>Deep Learning</i> e Redes Neurais	30
2.2.1	Tipos de Aprendizado	34
2.2.2	Erros	35
2.3	Metamodelos	39
3	COMPÓSITOS POLIMÉRICOS REFORÇADOS POR FIBRA DE CARBONO	43
3.1	Propriedades Mecânicas	44
4	METODOLOGIA	48
4.1	Configuração inicial	48
4.2	Modelo de célula unitária (RVE)	48
4.3	Caracterização linear do material	49
4.4	Caracterização não linear do material	52
4.5	Modelo de redes neurais	54
5	RESULTADOS E DISCUSSÃO	57
5.1	Resultados	57
6	CONCLUSÃO	71
	Referências	73
	Referências	73

APÊNDICES

78

APÊNDICE A – MÉTRICAS EM RELAÇÃO ÀS ÉPOCAS PARA TODAS AS REDES TREINADAS 79

1 Introdução

1.1 Materiais compósitos

Na área de Engenharia Estrutural e na Ciência e Engenharia de Materiais, a diferença entre uma estrutura e um material se dá pela presença de fronteiras. Um *material* é uma substância da qual um corpo é composto (ABOUDI; ARNOLD; BEDNARCYK, 2013). Por si só, o material não possui fronteiras, mas pode ser pensado como aquilo que compõe um ponto de um corpo. Cientistas e engenheiros desenvolveram diversas maneiras de descrever os materiais por meio de suas propriedades — módulo de Young, condutividade termal, densidade, por exemplo — assim descrevendo como um material se comporta em um corpo. Exemplificando, uma viga de aço (corpo), tem como estrutura a viga e como o material o aço.

Ainda de acordo com Aboudi, Arnold e Bednarczyk (2013), essa diferenciação entre material e estrutura é fundamental para delinear os objetivos e motivações de um determinado estudo. Deve-se ressaltar, no entanto, que ambos são importantes: por exemplo, ao dobrar uma viga, a forma com que ela se dobra é relacionada à rigidez flexional, e é proporcional multiplicação do módulo de Young, também conhecido como módulo elástico ou de elasticidade, pelo momento de inércia do corte transversal. Em outras palavras, embora o módulo de Young seja uma propriedade do material e o módulo de inércia seja uma propriedade estrutural, ambos combinados resultam em um comportamento específico do corpo. Assim, a separação entre propriedades estruturais e materiais facilita a abordagem de cada situação, pois todos os corpos feitos exatamente do mesmo material possuirão, por exemplo, o mesmo módulo de Young, enquanto cada estrutura diferente terá um momento de inércia diferente, resultando em diferentes padrões de dobras para cada corpo.

Essa discussão implica que um material é algo contínuo e que preenche completamente o espaço que ocupa. Dessa forma o material pode ser modelado utilizando-se da mecânica do contínuo, não possuindo detalhes internos como a presença de inclusões, orientação de grãos, ou arranjo molecular (ABOUDI; ARNOLD; BEDNARCYK, 2013).

Por outro lado, tanto no caso de materiais quanto no de estruturas, existem os *compósitos*, constituídos de duas ou mais entidades. Assim, uma *estrutura de compósito* é qualquer corpo feito de duas ou mais partes ou materiais. Já um *material compósito* é constituído de dois ou mais materiais diferentes com uma interface reconhecível entre eles. Mesmo recebendo a denominação de compósito, ainda é considerado um material, e portanto não possui fronteiras externas. No caso de se introduzir uma fronteira externa, torna-se uma estrutura composta de materiais compósitos, o que é um tipo particular de estrutura de compósito.

Para estudar um compósito, pode-se utilizar de diversas abordagens. No caso de se ignorar

as fronteiras internas do material, a mecânica do contínuo pode ser aplicada, considerando o material como anisotrópico e pseudo-homogêneo tornando-se assim dependente de propriedades “efetivas” ou “homogeneizadas”. Caso utilize-se da micromecânica, as fronteiras devem ser consideradas, assim como seus efeitos no material. Nesse caso, assume-se que cada material componente individual como algo contínuo, porém com suas propriedades representativas únicas e seu arranjo ditando o comportamento do material compósito como um todo.

Os compósitos são construídos a partir de uma ou mais *fases de reforço*, embutidas em uma outra fase chamada de *matriz*. De acordo com Kaw (2007), enquanto as matrizes são em geral contínuas, os reforços são diversos, podendo ser na forma de fibras, partículas ou flocos, por exemplo. Em geral, na descrição de um compósito são indicadas as frações dos volumes de cada fase presente. Para compósitos de duas fases, como por exemplo os reforçados por fibra, apenas a fração do volume ocupada pelo reforço é mostrada, sendo que o volume da matriz é dado por $v_m = 1 - v_f$ já que a soma de todas as fases deve ser 100 %, supondo-se uma porcentagem desprezível de vazios na mistura (ABOUDI; ARNOLD; BEDNARCYK, 2013).

O motivo de se utilizar materiais compósitos é que metais monolíticos e suas ligas nem sempre alcançam as demandas das tecnologias atuais. Combinando diversos materiais torna-se possível alcançar os requisitos necessários de desempenho. Por exemplo, em satélites as treliças precisam ser dimensionalmente estáveis no espaço, quando sujeitas a grandes variações de temperaturas, como de $-160,0\text{ °C}$ à $93,3\text{ °C}$. Materiais monolíticos não conseguem alcançar coeficientes de expansão térmica tão baixos, ao contrário dos compósitos de grafite/epóxi (KAW, 2007).

Em um geral, os materiais compósitos serão mais eficientes em diversas aplicações, quando comparados aos convencionais, podendo aumentar a resistência ou a rigidez, diminuir a fadiga mecânica, aprimorar a resistência a impactos, aumentar a condutividade térmica e a resistência à corrosão, para citar algumas possibilidades. Citando um exemplo prático, atualmente a indústria aeronáutica investe na diminuição da massa das aeronaves sem reduzir sua rigidez ou sua resistência, e para tal é possível trocar as ligas metálicas por materiais compósitos. Mesmo com custo mais alto, a redução no número de peças no conjunto e a diminuição do combustível gasto podem fazer esses materiais economicamente mais atraentes (KAW, 2007).

No entanto, também existem desvantagens e limitações no uso de compósitos, como por exemplo seu alto custo de fabricação. Segundo Kaw (2007), o custo de fazer uma estrutura de um compósito de grafite/epóxi pode ser de 10 a 15 vezes o custo de um material convencional. Melhorias nas tecnologias de processo e manufatura podem diminuir esses custos no futuro. Outro problema recorrente é o da caracterização mecânica de uma estrutura de compósito ser muito mais complexa que a de uma estrutura metálica. Isso ocorre pois os compósitos são geralmente anisotrópicos, ou seja, suas propriedades não são iguais em todas as direções, resultando em um maior número de parâmetros, tornando a análise estrutural tanto computacionalmente quanto experimentalmente custosa (KAW, 2007).

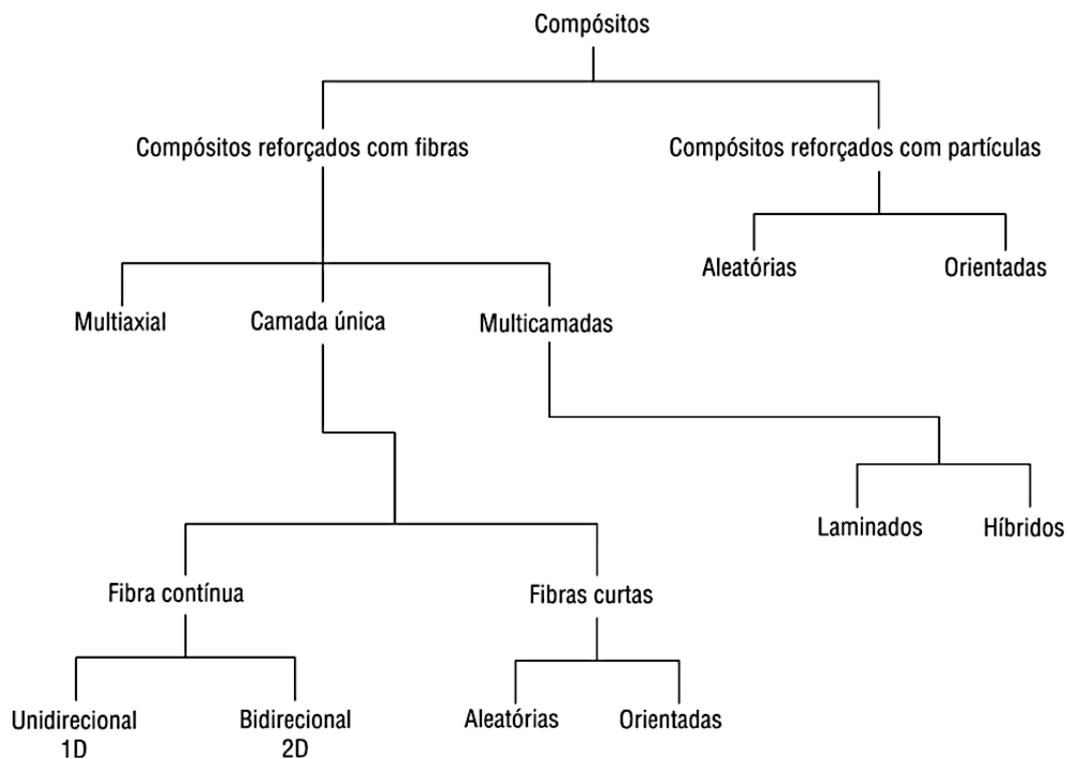
Outros fatores negativos que podem ser elencados são o processo de reparo complexo, a baixa combinação entre resistência geral e a fraturas, quando comparados aos metais, e que não há garantias de que compósitos necessariamente terão um desempenho melhor em todas as propriedades usadas na seleção de materiais (KAW, 2007).

Nas aplicações em indústrias como aeronáutica, aeroespacial e biomédica, o desempenho estrutural é indispensável, sendo que os fatores econômicos acabam tendo importância secundária. Já em indústrias como automobilística e construção civil, o parâmetro mais desejado é o baixo custo. Esses tipos de diferenças mostram como os materiais compósitos podem ser muito úteis na indústria de maneira geral (NETO; PARDINI, 2016).

1.2 Caracterização

Os compósitos podem ser divididos em dois grupos quanto a sua origem, naturais ou sintéticos, ou também devido ao seu reforço, podendo esse ser na forma de fibras, partículas ou flocos (KAW, 2007; NETO; PARDINI, 2016). A partir de combinações de diferentes matrizes e reforços, obtêm-se efeitos sinérgicos que geram propriedades que os materiais não apresentariam separadamente (NETO; PARDINI, 2016; MOURITZ, 2012). Essas combinações podem ser apreciadas na Figura 1, que organiza os compósitos de maneira geral por seu tipo de reforço.

Figura 1 – Esquema ilustrativo das diferentes classificações de um material compósito, a partir das possíveis combinações de reforço, número de camadas e tipo de fibra.



Fonte: Adaptado de Neto e Pardini (2016, p. 22).

Dentre os compósitos naturais, pode-se citar a madeira e o osso. A madeira tem a matriz formada por lignina e é reforçada por fibras de celulose; já o osso é formado por fibras de colágeno macio e placas de sal de osso feitas de íons de cálcio e fosfato como reforço (KAW, 2007).

Os reforços ainda podem ser naturais, mesmo com uma matriz sintética, como fibra de juta, coco e de sisal. No geral essas fibras são materiais lignocelulósicos formados fundamentalmente por celulose, hemicelulose e lignina, sendo amplamente utilizados como reforço em compósitos poliméricos por conta de possuírem baixos custo e massa específica, e uma boa facilidade de processamento. As fibras vegetais ainda possuem alta resistência e módulo elástico, no entanto têm baixa resistência térmica e a degradação mecânica em processamentos com alto grau de cisalhamento, como em extrusão e injeção. Essas fibras são mais utilizadas na indústria têxtil, de papel, ou combustível (LORANDI; CIOFFI; JUNIOR, 2016).

Os compósitos reforçados por partículas geralmente são materiais isotrópicos devido à forma aleatória que estas são adicionadas em matrizes como ligas e cerâmicas. Um exemplo desse tipo de compósito é a junção de cascalho, areia e cimento para fazer concreto. No caso de reforços em flocos, estes são introduzidos na forma de planos nas matrizes, geralmente sendo feitos de vidro, alumínio ou prata. Mesmo apresentando algumas vantagens, os flocos são difíceis de serem orientados, e apenas um número limitado de materiais estão disponíveis pra uso (KAW, 2007).

Existem dois tipos distintos de compósitos sintéticos reforçados por fibra, foco desse trabalho: aqueles reforçados por fibras curtas, logo descontínuas, e os reforçados por fibras longas, logo contínuas. Geralmente as fibras são anisotrópicas e feitas de materiais como carbono e aramida, e as matrizes podem ser feitas de materiais como epóxi, metais e cerâmicas. O compósito reforçado por fibra contínua pode ser caracterizado por sua unidade fundamental, que são lâminas de fibra empilhadas de forma unidirecional ou tecida (KAW, 2007).

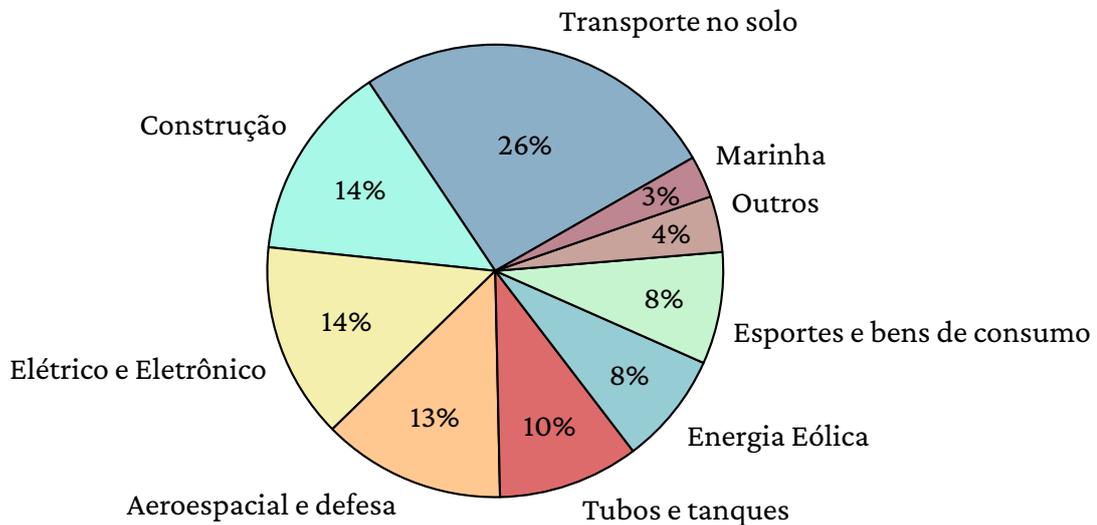
Os compósitos de uso industrial mais comuns são os denominados *compósitos de matriz polimérica* (PMCs, do Inglês, *polymer matrix composites*), onde um polímero compõe a matriz — por exemplo epóxi, poliésteres — reforçados por fibras finas, de grafite ou vidro (KAW, 2007). Apesar do melhor desempenho estrutural comparado ao dos compósitos naturais, o uso de matérias-primas vegetais vem crescendo nos últimos anos (NETO; PARDINI, 2016) devido à urgente necessidade de preservação do meio ambiente e minimização da dependência de recursos não renováveis.

1.3 Aplicações

O desenvolvimento de materiais compósitos de alto desempenho começou por volta de 1940 com a introdução de compósitos de matriz polimérica reforçados por fibra de vidro. O desenvolvimento continuou com a introdução de outros sistemas compósitos sejam eles cerâmicos, poliméricos ou metálicos, tornando-se uma grande força no mercado de materiais. Atualmente esse tipo de material está presente em diversas áreas como a aeroespacial, automotiva, construção,

esportes e lazer, e até mesmo na área de energia eólica. O mercado global de compósitos pode ser dividido entre os setores da Figura 2

Figura 2 – Divisão percentual do uso de compósitos por setor produtivo.



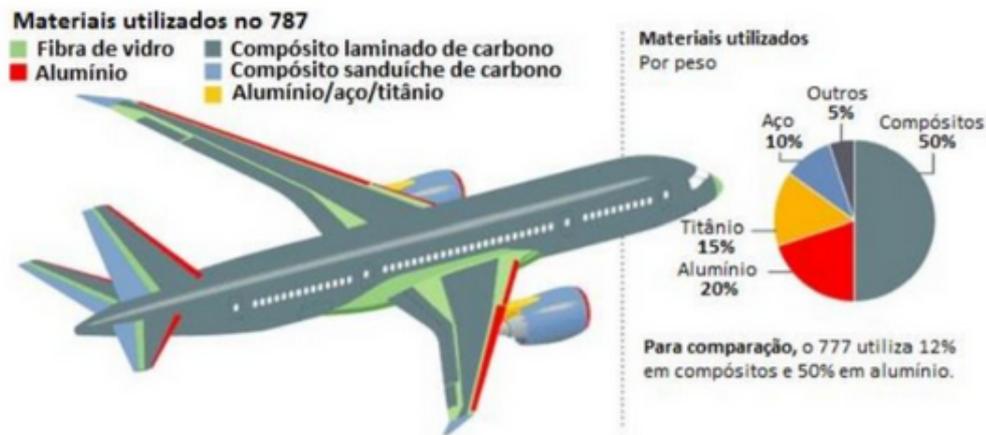
Fonte: Adaptado de Bunsell, Joannès e Thionnet (2021).

Pode-se dizer que a indústria que estuda e utiliza os compósitos de maneira mais expressiva é a aeroespacial, pois estes aparecem em diversos componentes estruturais como na fuselagem, asas, elevadores, estabilizadores, lemes, ailerons, portas, rotores e tanques. Isso ocorre pois, como dito anteriormente, a propriedade de reduzir o peso da aeronave é muito atrativa. O uso de alguns compósitos, como os de matrizes poliméricas, no entanto, é restrito a partes que operem em temperaturas de até 150 °C, para evitar amolecimento ou distorção por calor. As fibras de carbono laminadas de matriz polimérica de alta temperatura, como bismaleimida, são geralmente os compósitos escolhidos para motores a jato, no entanto, até essas são limitadas em temperatura de operação, sendo por volta de 200 °C-220 °C. Essa também foi a primeira indústria a utilizar os compósitos reforçados por fibra (MOURITZ, 2012; BUNSELL; JOANNÈS; THIONNET, 2021).

Uma demonstração do uso desse tipo de material nessa indústria é o Boeing 787, que possui 50 % de sua massa feita de compósitos, como pode ser apreciado na Figura 3. Só com a aplicação de compósitos de fibra de carbono nos dois motores GENx dessa aeronave já se conseguiu uma redução de cerca de 350 kg de peso, o que implica diretamente em economia de combustível e maior autonomia de voo (MOURITZ, 2012).

Um outro tipo de compósito que vem sendo pesquisado para uso em diversas áreas são os de nanotubos de carbono. Esses vêm sendo estudados para serem aplicados em sistemas mecânicos, de armazenamento de energia, de sensoriamento, e de emissão de campo. Na área aeroespacial, principalmente, os compósitos poliméricos com nanotubos de carbono como reforço são considerados a vanguarda do futuro, pois podem desempenhar um papel revolucionário. No entanto, ainda existem alguns desafios para a aplicação mais ampla desse material, como a

Figura 3 – Presença percentual dos diversos tipos de materiais na estrutura dos aviões Boeing A787. Os compósitos respondem por 50 % do total.



Fonte: Conejo (2019).

dispersão uniforme na estrutura e o método de produção a larga escala, entre outros (KAUSAR; RAFIQUE; MUHAMMAD, 2017).

O problema da dispersão do nano reforço pela matriz polimérica é um dos principais obstáculos a ser superado para obter-se uma aplicação satisfatória dos nanotubos de carbono como reforço nano estrutural. Devido às interações de van der Waals e à grande área superficial dos nanotubos, o material normalmente fica na forma de pequenos aglomerados, o que é influi negativamente nas propriedades mecânicas do compósito final. Por conta da elevada razão de aspecto dos nanotubos, a capacidade de processamento dos compósitos também é dificultada, pois essa é responsável por um aumento significativo da viscosidade do polímero, o que dificulta o processo de dispersão. Por tais motivos, a utilização de nanotubos é limitada ainda em cerca de 5 % do volume em matriz polimérica (CONEJO, 2019).

1.4 Modelagem matemática

No contexto da modelagem de compósitos, ou seja, nas simulações e análises do comportamento da estrutura ou do material, existem duas abordagens básicas: a macromecânica e a micromecânica (MOURITZ, 2012).

A abordagem *macromecânica* trata de analisar o material considerando apenas a macro-escala, ou seja, considera-se o compósito como algo anisotrópico, ao mesmo tempo em que são ignorados os detalhes sobre a configuração dos materiais constituintes. Tratando-se de elasticidade linear, ou seja, assumindo uma relação linear entre os componentes de tensão e deformação, essa abordagem é direta, utilizando métodos como elementos finitos (HUGHES, 2012) para determinar parâmetros do material ou seu desempenho estrutural; esse é o procedimento padrão para o desenvolvimento de estruturas de compósitos. No entanto, mesmo sendo computacional-

mente eficiente e direta, essa abordagem depende fortemente de dados experimentais custosos, como a geração de cada variação diferente do compósito, como por exemplo diferenças na fração volumétrica da fibra (ABOUDI; ARNOLD; BEDNARCYK, 2013).

No regime não linear — aplicações em alta temperatura, por exemplo — a abordagem macromecânica pode ser ineficaz na previsão de dados sobre dano e falha, por exemplo. Por ser um material anisotrópico, os modelos de dano e falha devem ser construídos considerando toda a variedade de comportamento e mecanismos de falha do compósito em varias direções distintas, caracterizada por meio de extensos testes. Mesmo que muitos busquem um modelo não linear para o comportamento de compósitos na macromecânica, essa abordagem é sempre muito fenomenológica, ou seja, ligada ao que pode ser visto e analisado diretamente no material, o que por definição, não considera o que acontece em cada constituinte na escala física adequada (ABOUDI; ARNOLD; BEDNARCYK, 2013).

Do outro lado está a abordagem *micromecânica*, que considera todos os materiais constituintes para prever o comportamento efetivo e heterogêneo do material baseado no comportamento de cada constituinte individual, além de seu arranjo geométrico. Determinar o comportamento efetivo de um compósito por meio da micromecânica permite tratá-lo como um material em análises de escalas maiores, como a macromecânica. Nesta abordagem, as propriedades efetivas de um material compósito podem ser determinadas por meio da micromecânica e usadas numa análise de um laminado para representar as camadas dos materiais, ou até mesmo em elementos finitos de uma estrutura compósita para representar os diferentes materiais em cada região (ABOUDI; ARNOLD; BEDNARCYK, 2013).

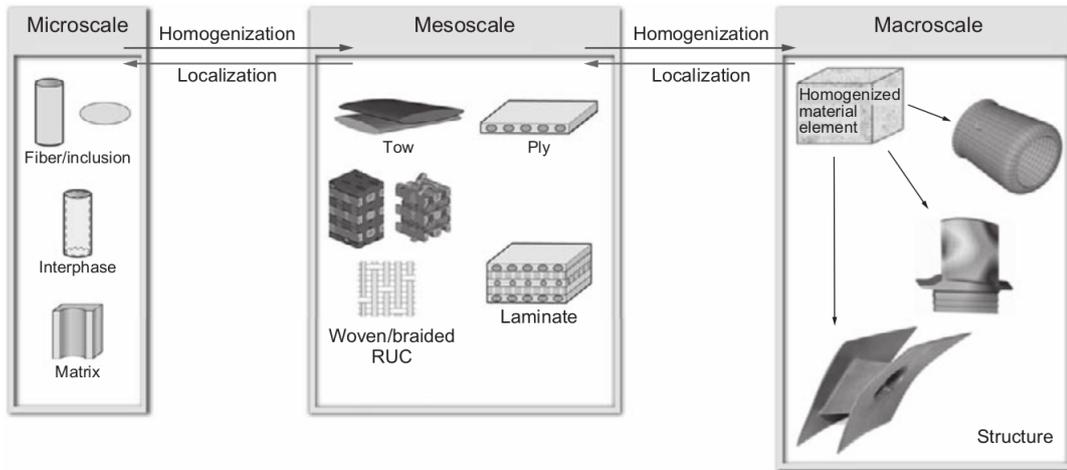
Um benefício da abordagem micromecânica é que as propriedades do compósito podem ser determinadas em qualquer direção, para qualquer fração volumétrica de fibra ou arquitetura do reforço, até mesmo se esse material nunca foi manufaturado (ABOUDI; ARNOLD; BEDNARCYK, 2013).

Uma forma de compatibilizar as dificuldades e as vantagens de ambas as abordagens é a *modelagem multiescala* (do Inglês, *multiscale modelling*), uma maneira de simular o comportamento de compósitos por meio de diversas escalas de tempo ou comprimento. Na Figura 4 mostramos como essas escalas interagem entre si.

Da esquerda para a direita na Figura 4 temos a *microescala*, que trata do nível constituinte (fibra, matriz, interface), a *mesoescala*, que trata da organização dos constituintes, e a *macroescala*, que trata da estrutura global. Normalmente, percorre-se essas escalas por meio de técnicas de homogeneização, usadas na obtenção das propriedades de um material heterogêneo a partir das leis constituintes e da distribuição espacial dos componentes, e de localização, que faz o processo inverso (ABOUDI; ARNOLD; BEDNARCYK, 2013; KANIT *et al.*, 2003).

Essa modelagem pode ser classificada entre *hierárquica*, *sinérgica* ou *concorrente* e existem diversos métodos para fazê-la, podendo ser *analíticos*, *semianalíticos*, ou *numéricos*. Todas essas

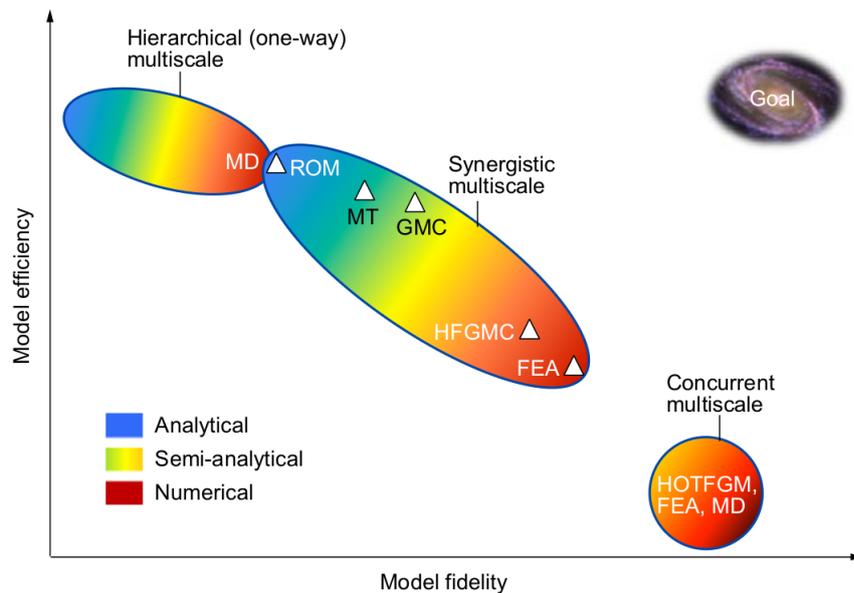
Figura 4 – Níveis de uma análise multiescala de materiais compósitos.



Fonte: Aboudi, Arnold e Bednarcyk (2013, p. 10).

possibilidades e suas eficiências e fidelidades são resumidas na Figura 5.

Figura 5 – Esquema ilustrativo do balanço entre eficiência e fidelidade de diferentes métodos da modelagem multiescala. Todas as siglas vêm do Inglês, sendo MD-*Molecular Dynamics*; ROM-*Rule of mixtures*; MT-*Mori-Tanaka*; GMC-*Generalized Method of Cells*; HFGMC-*High-Fidelity Generalized Method of Cells*; FEA-*Finite Elements Analysis*; HOTFGM-*High Order Theory for Functionally Graded Materials*.



Fonte: Aboudi, Arnold e Bednarcyk (2013, p. 13).

Para a modelagem com a finalidade de encontrar as propriedades macroscópicas de um compósito, normalmente faz-se uso da homogeneização por meio do método dos elementos finitos (PATHAN; PONNUSAMI *et al.*, 2019). Uma técnica muito utilizada na homogeneização é a do elemento representativo de volume (RVE, do Inglês *Representative Volume Element*) (PATHAN; TAGARI-

ELLI *et al.*, 2017), abordada em diversos trabalhos como os de (PATHAN; PONNUSAMI *et al.*, 2019; PATHAN; PATSIAS *et al.*, 2017; HADDEN *et al.*, 2015; UTZIG *et al.*, 2018; SONG; KRISHNASWAMY; PUCHA, 2016; WAN *et al.*, 2020).

Essa técnica consiste em encontrar um volume V de um material heterogêneo, como um compósito, que seja grande o suficiente para representar estatisticamente o material como um todo. No entanto, esse volume deve ser pequeno o suficiente para ser considerado um elemento de volume na mecânica contínua (KANIT *et al.*, 2003). Na literatura existem diversas definições de como quantificar o tamanho mínimo de um RVE; no entanto, esse valor depende do material que está sendo considerado e qual propriedade está sendo estudada (STROEVEN; ASKES; SLUYS, 2004; KANIT *et al.*, 2003; PATHAN; TAGARIELLI *et al.*, 2017).

Recentemente, além da utilização da simulação computacional clássica, trabalhos sugerem propostas da utilização de Inteligência Artificial para a modelagem de compósitos reforçados por fibra. Neste sentido, o *Aprendizado de Máquina* já está sendo explorado, principalmente por conta da vantagem na rapidez de obtenção de resultados e do custo computacional mais baixo, permitindo explorar estratégias de projetos que obtenham compósitos de alto desempenho (CHEN; GU, 2019). Trabalhos que se utilizam de *Aprendizado de Máquina* chegam a diminuir o esforço computacional em mais de 6 ordens de magnitude, por exemplo, ao buscar prever a resposta multiaxial de detecção de deformação de compósitos poliméricos com nanotubos de carbono (MATOS; PINHO; TAGARIELLI, 2019a).

Já existem trabalhos, por exemplo, com uma comparação da desempenho do aprendizado de máquina para otimização do projeto de compósitos laminados de rigidez variável (ARIAN NIK *et al.*, 2014), usados para prever propriedades de compósitos poliméricos reforçados por fibra de carbono (QI *et al.*, 2019) e que propõem o uso de redes neurais artificiais para determinar a condutividade elétrica de compósitos poliméricos reforçados por nanotubos de carbono (MATOS; PINHO; TAGARIELLI, 2019b).

O objetivo nesse trabalho é o de simular uma propriedade mecânica de materiais compósitos reforçados por fibra utilizando-se de redes neurais, uma das várias técnicas de aprendizado de máquina. Mais especificamente, utilizar de uma rede neural para modelar um compósito carbono-epóxi para prever o módulo de Young em diferentes configurações. Para tal, os Capítulos 2 e 3 dedicam-se respectivamente à introdução do que são o aprendizado de máquina e o módulo de Young. Em seguida será apresentada a metodologia utilizada no Capítulo 4, e por fim os resultados no Capítulo 5, a conclusão no Capítulo 6, e finalmente as referências utilizadas ao longo do trabalho. Tem-se ainda o Apêndice A, onde são apresentados os gráficos de treinamento das redes avaliadas na execução do trabalho.

2 Aprendizado de Máquina

2.1 Contexto histórico

Desde a Grécia antiga existem figuras mitológicas que representam o papel de inventores que em suas histórias criam máquinas capazes de pensar, como Talos, autômato criado por Hefesto que protegia a ilha de Creta. Esse sonho de criar máquinas que são capazes de pensar é algo que se manteve na humanidade, tanto que quando o primeiro computador programável foi concebido, as pessoas já se perguntavam sobre quando essas máquinas poderiam se tornar inteligentes (GOODFELLOW; BENGIO; COURVILLE, 2016).

Em Menabrea (1842), Lady Ada Lovelace, quando falava sobre o protótipo do *Analytical Engine* — primeiro computador mecânico de uso geral — diz que este não tinha pretensão de originar nada, apenas podia fazer qualquer coisa que soubéssemos como determiná-lo a fazer, sem a capacidade de antecipar qualquer relação analítica ou verdade. Já Turing (1950) introduziu o que seriam os conceitos do que seria uma *Inteligência Artificial (IA)*, ponderando sobre a afirmação de Lovelace, e questionando-se se esses computadores seriam capazes de aprender ou de originalidade, concluindo finalmente que sim, seria possível (CHOLLET, 2017).

2.1.1 O nascimento da IA

A IA então surge nesse contexto, entre os anos 1940-1950, quando diversos pioneiros da ciência da computação fizeram essas mesmas perguntas. A principal ideia dessa área é fazer com que tarefas intelectuais que normalmente são executadas por humanos, sejam automatizadas (GOODFELLOW; BENGIO; COURVILLE, 2016; CHOLLET, 2017). No início das pesquisas na área, diversos problemas difíceis de serem solucionados por humanos foram resolvidos de maneira direta por computadores, principalmente aqueles que podiam ser descritos por uma lista formal de regras matemáticas. No entanto, resolver tarefas que eram facilmente ou intuitivamente resolvidas por humanos, mostraram-se extremamente complexos para a inteligência artificial (GOODFELLOW; BENGIO; COURVILLE, 2016).

Um exemplo de tarefa que as máquinas conseguem tratar bem é o xadrez, sendo inclusive uma das tarefas propostas por Turing (1950) para competir com humanos. As máquinas desenvolvidas para isso eram explicitamente programadas, utilizando-se de um conjunto de regras fixas para manipular o conhecimento. Esse tipo de abordagem é chamada de IA simbólica, e foi um paradigma dominante entre os anos 1950 e 1980, tendo seu pico de popularidade com os sistemas especialistas por volta dos anos 1980. Em 1997, uma máquina da IBM chamada *Deep Blue*, conseguiu vencer o campeão mundial da época Garry Kasparov (CHOLLET, 2017; GOODFELLOW; BENGIO; COURVILLE, 2016). No entanto, por existirem 10^{52} possíveis estados do sistema num

tabuleiro de xadrez, obter uma IA que obtenha sempre o melhor resultado possível é até hoje um desafio computacional imenso (MAHARAJ; POLSON; TURK, 2022).

Mesmo a IA simbólica funcionando bem em problemas lógicos e bem definidos como o xadrez, essa não apresenta um bom desempenho descobrindo regras explícitas para resolver problemas mais complexos, como problemas de lógica difusa, classificação de imagens, reconhecimento de fala e tradução de línguas, entre outros. É nesse vácuo que surge o *Aprendizado de Máquina* (CHOLLET, 2017).

Um exemplo dessa dificuldade em resolver problemas intuitivos para humanos é o do projeto Cyc, um sistema que se utilizava de uma base de dados de afirmações em uma linguagem chamada Cycl (GUHA; LENAT, 1991). Dada a dificuldade inerente à criação de regras formais que consigam descrever a complexidade do mundo, a máquina teve problemas na interpretação de uma história sobre uma pessoa chamada Fred que se barbeava pela manhã. Isso ocorria pois a máquina entendia que pessoas não possuíam partes elétricas, no entanto, Fred estava com um barbeador elétrico, logo durante o ato de se barbear, Fred possuía partes elétricas. Essa lógica levava ao questionamento de se ele continuava humano enquanto se barbeava (GOODFELLOW; BENGIO; COURVILLE, 2016).

As barreiras enfrentadas por esses sistemas de código com regras explicitamente programadas mostram que as IAs precisam da habilidade de adquirir seu próprio conhecimento, podendo obtê-lo através do reconhecimento de padrões. Essa habilidade foi denominada Aprendizado de Máquina (GOODFELLOW; BENGIO; COURVILLE, 2016).

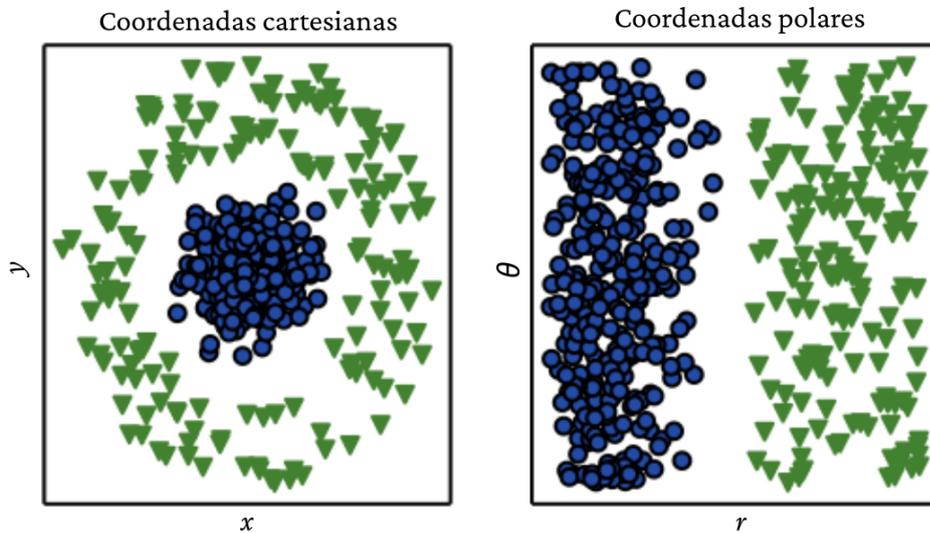
2.1.2 Aprendizado de Máquina

O Aprendizado de Máquina surge de perguntas ainda relacionadas à afirmação de Lovelace: pode um computador ir além daquilo que sabemos como determinar fazer? Como ele pode, de maneira autônoma, realizar uma determinada tarefa? Um computador pode nos surpreender? Ao invés de programadores descobrirem regras a partir dos dados disponíveis, o computador poderia aprendê-las apenas “olhando” para os dados (CHOLLET, 2017)?

Essas perguntas levaram a um novo paradigma na computação. Na programação clássica, usado no paradigma da IA simbólica, o ser humano era responsável por colocar as regras e os dados de entrada, e assim eram obtidas as respostas na saída. No Aprendizado de Máquina o ser humano fornece os dados de entrada e as respostas de saída, e obtém as regras. Essas mesmas regras, automaticamente descobertas, podem ser reutilizadas em novos dados, produzindo novas respostas. Assim, pode-se dizer que um sistema de Aprendizado de Máquina é mais treinado, do que explicitamente programado (CHOLLET, 2017). Nesse novo paradigma, os computadores podem resolver problemas do mundo real que necessitem decisões aparentemente subjetivas com maior facilidade (GOODFELLOW; BENGIO; COURVILLE, 2016).

O problema central no Aprendizado de Máquina é o de transformar os dados significativa-

Figura 6 – Exemplo de diferentes representações de um mesmo conjunto de dados à serem separados em duas categorias por uma linha. À esquerda, o conjunto de dados em coordenadas cartesianas; à direita, o mesmo conjunto de dados em coordenadas polares.



Fonte: Goodfellow, Bengio e Courville (2016, p. 4).

mente, ou seja aprender representações úteis dos dados de entrada, pois o desempenho desses algoritmos depende fortemente dessa representação. Cada parte da informação é encarada como uma característica, e a máquina aprende como cada uma dessas características se relaciona com os diversos resultados. Um exemplo da importância na representação dos dados pode ser demonstrada é o mesmo conjunto de dados em coordenadas cartesianas ou em coordenadas polares, como ilustra a Figura 6.

Segundo Chollet (2017), atualmente todos os algoritmos de aprendizado consistem em encontrar automaticamente estas transformações que tornam os dados em uma representação mais útil para determinada característica. Via de regra os algoritmos não são criativos para encontrar as transformações, apenas procuram em um conjunto predefinido de operações, chamado *espaço de hipóteses*.

Assim, pode-se dizer que o Aprendizado de Máquina trata do problema de encontrar uma representação para um conjunto de dados de entrada, dentro de um espaço predefinido de possibilidades, utilizando um sinal de retorno como guia (CHOLLET, 2017).

Diversos problemas podem ser resolvidos a partir da definição de um conjunto de características a serem extraídas desse problema e provendo essas características para um simples algoritmo de aprendizado. Um dos objetivos do programador é separar adequadamente os fatores de variação dos dados observados, para assim determinar quais são as características que serão aprendidas pela máquina. No entanto esse é um trabalho de alta dificuldade, pois muitos desses fatores podem ser de difícil identificação (GOODFELLOW; BENGIO; COURVILLE, 2016).

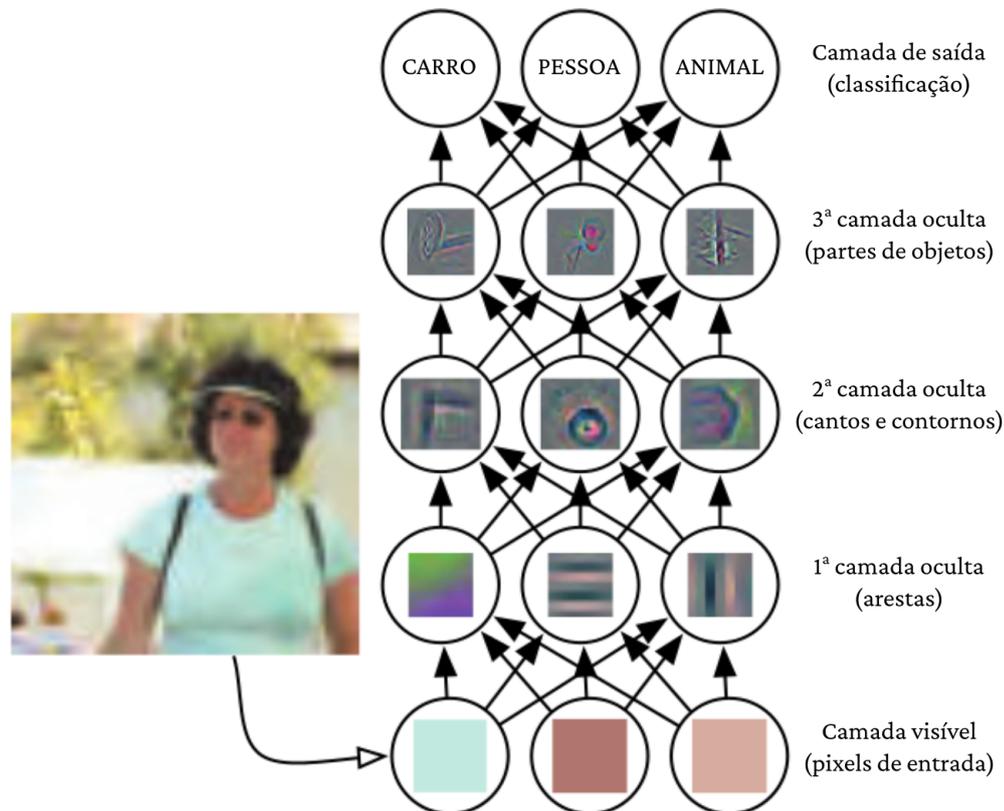
2.1.3 Aprendizado Profundo

Uma forma de abordar o problema de representação dos dados é o que hoje é chamado de *Deep Learning* (do Inglês, *Aprendizado Profundo*), pois esse permite com que o computador crie conceitos mais complexos (GOODFELLOW; BENGIO; COURVILLE, 2016). O *Deep Learning* é um subcampo específico do Aprendizado de Máquina que possui uma visão diferente do aprendizado de representações a partir de dados, e que foca o aprendizado na construção de camadas sucessivas de representações cada vez mais significativas. Em outras palavras, o *deep* de *Deep Learning* trata da profundidade do modelo, ou seja, de quantas camadas contribuem. Atualmente, esse tipo de aprendizado utiliza de dezenas e até centenas de camadas sucessivas de representação, todas aprendendo automaticamente pela exposição aos dados de treino. Enquanto isso, outras abordagens que tendem a utilizar apenas uma ou duas camadas de representações de dados são chamadas de *shallow learning* (do Inglês, *Aprendizado Raso*) (CHOLLET, 2017).

Um exemplo comum de *Deep Learning* é uma rede densa, também chamada de *perceptron multicamadas*. Essa rede é uma função matemática criada pela composição de funções mais simples, que mapeia os valores de entrada aos de saída. No *Deep Learning* são criadas diversas camadas, que podem ser interpretadas como uma memória do computador depois de executar uma série de instruções em paralelo. Quanto maior o número de camadas, maior a profundidade da rede, e portanto mais instruções em sequência podem ser feitas. Essas instruções em paralelo garantem maior poder computacional, pois cada instrução seguinte pode se referir a resultados das instruções anteriores. A Figura 7 exemplifica uma rede densa com 1 camada visível, assim chamada pois as variáveis que nela estão são observáveis, e 3 camadas ocultas, cujo nome indica que os resultados dessas camadas não são diretamente acessáveis. A primeira camada é responsável por interpretar as bordas, a segunda por interpretar os cantos e contornos, e a terceira por interpretar partes de objetos (GOODFELLOW; BENGIO; COURVILLE, 2016).

Historicamente, os estudos na área de *Deep Learning* surgem por volta dos anos 1940; no entanto, os nomes foram sendo alterados ao longo do tempo. Entre 1940-1960, essa área era chamada de cibernética; entre 1980-1990, conectismo era o nome mais utilizado; por volta de 2006 o termo *Deep Learning* aparece com maior presença. Hoje em dia também é possível encontrar a mesma área como *Redes Neurais Artificiais* (ANNs, do Inglês *Artificial Neural Networks*) pois buscava-se fazer com que os modelos computacionais tivessem um aprendizado similar ao de um cérebro biológico (associado à busca por um melhor entendimento do próprio cérebro humano). No entanto, hoje a Neurociência não é mais o principal guia nessa área de estudos, devido à dificuldade de entender o cérebro na medida necessária para replicar suas funcionalidades em uma máquina (GOODFELLOW; BENGIO; COURVILLE, 2016).

As camadas, como as da Figura 7, são aprendidas a partir de modelos chamados *redes neurais*. O termo “redes neurais”, como já dito, é uma referência à neurobiologia, mas mesmo que alguns conceitos centrais do *Deep Learning* tenham inspirações no entendimento do cérebro, eles não são modelos do cérebro. Para propósitos gerais, *Deep Learning* é uma estrutura matemática

Figura 7 – Exemplo de um modelo de *Deep Learning* de reconhecimento de imagem.

Fonte: Goodfellow, Bengio e Courville (2016, p. 6).

para aprendizado de representações a partir de dados (CHOLLET, 2017).

Os primeiros modelos de *Deep Learning* eram modelos lineares simples motivados por uma perspectiva neurocientífica. Esses modelos usavam um conjunto de n variáveis de entrada $\mathbf{x} = (x_1, \dots, x_n)$ e as associava aos dados de saída y . Esses modelos aprenderiam por meio de uma série de pesos $\mathbf{w} = (w_1, \dots, w_n)$, tendo sua saída dada por uma função $f(\mathbf{x}, \mathbf{w}) = x_1w_1 + \dots + x_nw_n$ (GOODFELLOW; BENGIO; COURVILLE, 2016).

Após esse modelo, foram criadas novas funções nesse mesmo molde, além de outras baseadas no funcionamento do cérebro, como por exemplo o *neurônio de McCulloch-Pitts* que é capaz de reconhecer dois tipos de categoria nos dados de entrada; o *perceptron*, que é capaz de aprender os pesos que definirão categorias dados exemplos de entrada de cada uma delas; e o *ADALINE* (do Inglês, *adaptive linear element*), capaz de retornar o valor de $f(x)$ para prever um número real, também conseguindo aprender a prever esses números a partir de dados (GOODFELLOW; BENGIO; COURVILLE, 2016).

A Neurociência, mesmo não sendo mais o principal norteador, deixou diversos caminhos a serem explorados para a evolução das redes neurais. A ideia básica de que a interação de diversas unidades computacionais possam se tornar inteligentes vem diretamente do cérebro e seus neurônios. O *neocognitron* foi um modelo de arquitetura poderoso para processamento de imagem, e foi inspirado na estrutura do sistema visual dos mamíferos, tornando-se posteriormente a base para

as redes neurais convolucionais modernas. Atualmente, a maioria das redes neurais é baseada no modelo de neurônio chamado *unidade linear retificada* ou *ReLU* (do Inglês *rectified linear unit*). Os neurônios reais funcionam de forma muito diferente do que os neurônios *ReLU*, no entanto maior realismo neural não necessariamente leva a um melhor desempenho do aprendizado de máquina (GOODFELLOW; BENGIO; COURVILLE, 2016).

Mesmo que a mídia ainda enfatize muito a similaridade do *Deep Learning* com o cérebro e apesar da clara inspiração na Neurociência, os estudos atuais nessa área têm inspiração mais focada em outras áreas, especialmente nas relacionadas aos fundamentos da Matemática Aplicada, como Álgebra, Teoria de Probabilidades, Teoria da Informação e Otimização Numérica. No entanto, mesmo que a pesquisa dentro do *Deep Learning* não tenha mais seu foco em entender como o cérebro funciona em um nível algorítmico, essa área ainda existe, sendo hoje chamada de neurociência computacional (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.1.4 Conectismo

A ideia geral do movimento chamado *conectismo* (do Inglês, *connectionism*) era de que um grande número de unidades computacionais simples poderiam gerar um comportamento inteligente quando ligados. Essa ideia baseava-se na premissa de que um maior número de neurônios conectados levaria a uma melhor resposta da rede. E mesmo que alguns dos conceitos desse movimento estivessem errados, alguns permanecem até hoje (GOODFELLOW; BENGIO; COURVILLE, 2016).

Dentre eles o da *representação distribuída*, onde cada entrada no sistema deve ser representada por diferentes características, e essas devem estar envolvidas na representação de possíveis entradas. Por exemplo, num sistema de visão capaz de reconhecer carros, caminhões e pássaros, e esses objetos sejam vermelhos, verdes ou azuis. Uma maneira de representar os dados de entrada seria separar um neurônio para cada uma das nove possíveis combinações. Este fato requer nove diferentes neurônios e cada um deles deveria aprender de maneira independente o conceito de cor e a identidade do objeto. Uma maneira de melhorar esse exemplo é utilizar a representação distribuída, com três neurônios descrevendo cores e mais três neurônios identificando os objetos. Assim, são necessários apenas seis neurônios especializados: o neurônio responsável por identificar o carro consegue identificar qualquer cor de carro, ao invés de uma cor específica (GOODFELLOW; BENGIO; COURVILLE, 2016).

Outro conceito proveniente do movimento conectista, e que ainda hoje é extremamente relevante, é a utilização de *back-propagation* (em tradução livre do Inglês, *retro propagação*) para treinar redes neurais densas (GOODFELLOW; BENGIO; COURVILLE, 2016), como será detalhada mais adiante no trabalho.

Essa onda dos estudos de redes neurais durou até o meio dos anos 1990. As expectativas do mercado tornaram-se irreais, e os investidores se desapontaram com o descompasso entre os

resultados prometidos e os efetivamente alcançados. Enquanto isso, outras áreas do Aprendizado de Máquina faziam avanços com bons resultados em diversas tarefas. Tal fato diminuiu a popularidade das redes neurais até 2007. Até aquele momento acreditava-se que as redes eram muito difíceis de serem treinadas. No entanto, eventualmente percebeu-se que esse não era um problema intrínseco do algoritmo ou da estrutura da rede, mas sim do custo computacional, muito alto para as máquinas disponíveis à época (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.1.5 *Deep Belief Networks*

A terceira onda de pesquisa na área começa em 2006 com o trabalho pioneiro de Geoffrey Hinton, que mostrou que um tipo de rede neural chamada *deep belief network* (em tradução livre do Inglês, *rede de crença profunda*) podia ser treinada de maneira eficiente utilizando-se de um método chamado *greedy layer-wise pretraining* (em tradução livre do Inglês, *pré-treino guloso por camadas*). A maneira encontrada por Hinton, Osindero e Teh (2006) foi a de fixar os pesos das camadas mais distantes enquanto os pesos das primeiras camadas são aprendidos. Assim que uma camada aprende, seu peso é separado dos pesos das camadas seguintes, e uma nova camada pode aprender.

A partir desse estudo, outros grupos de pesquisadores mostraram que essa estratégia funcionava para diversos outros tipos de redes, e sistematicamente ajudaram a melhorar a capacidade de generalização dos algoritmos. Esse fato, somado a um aumento substancial no poder computacional, fez com que nesse período o termo *Deep Learning* começasse a se popularizar, já que os pesquisadores podiam treinar redes bem maiores (mais profundas). Essa onda dura até a atualidade, com uma mudança de paradigma: no princípio o foco era na habilidade dos modelos generalizarem bem pequenas bases de dados não categorizados. Hoje, graças ao enorme poder computacional disponível, a pesquisa e o desenvolvimento de algoritmos e modelos em *Deep Learning* se concentram em bases de dados enormes, porém categorizadas (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.2 *Deep Learning* e Redes Neurais

Modelos de *Deep Learning* aprendem a partir da combinação de um conjunto de funções, ponderadas por pesos w . Estes pesos parametrizam a transformação feita nos dados de entrada — doravante denotados por x — em cada camada, podendo então ser nomeados de parâmetros das camadas. Nesse contexto, aprender resume-se a encontrar o melhor conjunto de pesos para todas as camadas do modelo, permitindo construir um mapeamento entre entradas e saídas, denotadas por y , efetivamente generalizando a relação entre os dois conjuntos. Esse é um dos principais pontos do *Deep Learning*: um modelo pode possuir dezenas de milhões de parâmetros. Encontrar o melhor valor para cada um deles é uma tarefa computacionalmente intensa, pois cada um dos parâmetros influencia no comportamento de todos os outros (CHOLLET, 2017).

A medida de quão próximo o algoritmo está de uma boa generalização dos conjuntos de entrada e saída se dá através da chamada *função de perda* (ou função custo, ou ainda função de erro), denotada por $J(y, \hat{y})$, que quantifica a diferença entre a saída real, y e a previsão obtida pela rede, \hat{y} . A escolha da função de perda dependerá do tipo de problema em questão. Dentre as mais comuns temos o erro quadrático médio, a função de Huber, a entropia cruzada e a entropia cruzada binária (CHOLLET, 2017; BURKOV, 2019).

Uma rede neural artificial usual baseia-se no fato de que uma variável de entrada x é propagada para as unidades ocultas em cada camada, produzindo uma resposta \hat{y} . Esse modelo é chamado de *forward propagation* (em tradução livre do Inglês, *propagação direta*). Durante o treinamento, essa maneira de propagar a informação ocorre até produzir um custo escalar $J(\theta)$, sendo J um critério que expressa a preferência para os pesos distribuídos pela rede. O algoritmo de *back-propagation* permite que a informação possa voltar pela rede, com o propósito de computar o gradiente (GOODFELLOW; BENGIO; COURVILLE, 2016).

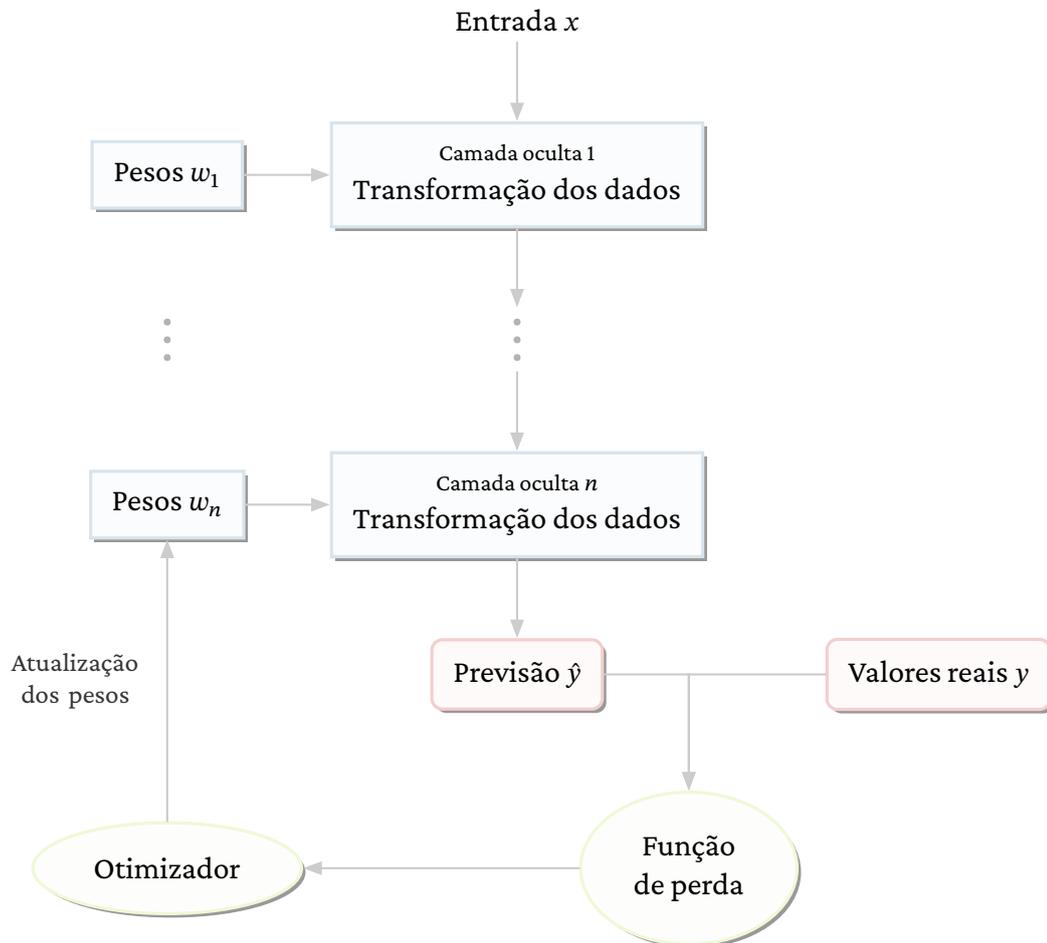
Inicialmente os pesos são escolhidos de forma aleatória, o que naturalmente produz um resultado longe do esperado. Mas devido ao emprego de *back-propagation* a rede otimiza seus pesos produzindo a queda na função de perda. Esse processo é chamado de *laço de treinamento*, e, repetido uma quantidade suficiente de vezes, resulta em pesos com valores que minimizam a função de perda. Um modelo com uma perda mínima é aquela que possui resultados tão próximos quanto o possível do esperado, e nesse ponto, diz-se que o modelo está treinado. Tal processo pode ser exemplificado pela Figura 8 (CHOLLET, 2017).

A *camada* é a estrutura de dados fundamental de uma rede neural. Essa é um módulo de processamento de dados que recebe um conjunto de dados — em geral representados por tensores — e devolve como resultado um ou mais tensores. Algumas camadas são ditas sem estado, significando que estas não possuem os pesos (CHOLLET, 2017).

Existem diferentes tipos de camadas, cada uma apropriada para diferentes formatos de tensores ou da natureza dos dados. Por exemplo, um dado vetorial simples, no formato de tensores bidimensionais, geralmente é processado por camadas densamente conectadas, ou apenas camadas densas. Já dados sequenciais, no formato de tensores tridimensionais, são geralmente processados por camadas recorrentes, como por exemplo, a camada LSTM (do Inglês, *Long-short Term Memory*). Já dados de imagens, em formato de tensores com quatro dimensões, geralmente são processados por camadas convolucionais bidimensionais (CHOLLET, 2017).

Cada camada pode ser encarada como um “tijolo” da rede neural: essa metáfora é possível pois um modelo de *Deep Learning* pode ser criado a partir da combinação de camadas compatíveis para gerar uma sequência de transformação de dados útil. A compatibilidade, nesse caso, refere-se ao fato de que cada tipo de camada aceita os tensores de entrada em um formato específico, e devolvem tensores em determinados formatos. Como menciona Chollet (2017), escolher a arquitetura certa de uma rede é mais uma arte do que uma ciência, mesmo que existam boas práticas e princípios a seguir.

Figura 8 – Diagrama ilustrativo do funcionamento de um modelo de *Deep Learning*. A camada de entrada recebe os dados de treinamento x , que serão transformados nas n camadas ocultas, ponderados pelos respectivos pesos $w_i, i = 1, \dots, n$. Na última cada será produzida uma previsão \hat{y} , que será comparada com os valores esperados y usando-se uma função de perda. O resultado dessa comparação é usado pelo otimizador para atualizar os pesos e reiniciar o processo.



Fonte: Adaptado de Chollet (2017, p. 11).

Cada camada é composta por unidades chamadas de *neurônios*. Cada unidade corresponde a uma dimensão no espaço de representação da camada, ou seja, possuir mais unidades permite que a rede aprenda representações cada vez mais complexas, porém além do aumento do custo computacional, pode levar ao aprendizado de padrões indesejados (CHOLLET, 2017).

As unidades utilizam *funções de ativação* para aumentar seu espaço de hipóteses, ou seja, para possuir uma maior gama de transformações possíveis. Uma rede com camadas densas sem uma função de ativação, por exemplo, consegue apenas realizar duas operações lineares, o produto interno dos pesos com a entrada, e uma soma, como mostrado na Equação (2.1)

$$\hat{y} = w \cdot x + b. \quad (2.1)$$

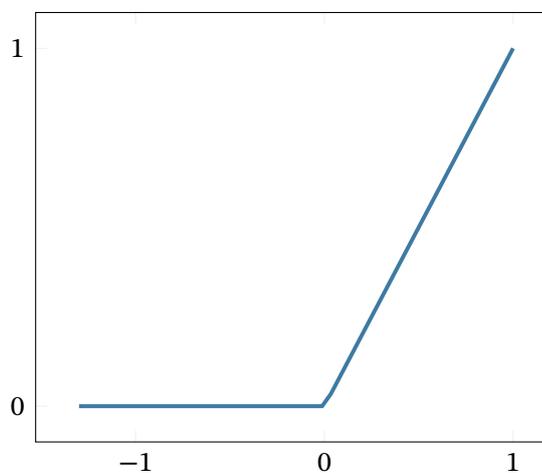
Assim, cada camada possui um espaço de hipóteses de todas as possíveis transformações lineares

dos dados de entrada em um espaço de dimensão igual ao número de unidades, o que torna seu escopo bastante limitado. Para gerar um espaço de hipóteses mais rico, utiliza-se da não-linearidade, oferecida pela função de ativação (CHOLLET, 2017).

Existem diversas funções de ativação (por exemplo, sigmoide, exponencial, SoftMax, tanh, para citar algumas das mais comuns), e entre elas a mais popular é a ReLU. Essa função pode ser expressa pela Equação (2.2) (GOODFELLOW; BENGIO; COURVILLE, 2016) e apreciada no gráfico da Figura 9.

$$g(z) = \max\{0, z\}. \quad (2.2)$$

Figura 9 – Função da unidade linear retificada, ReLU.



Fonte: Criada pelo autor.

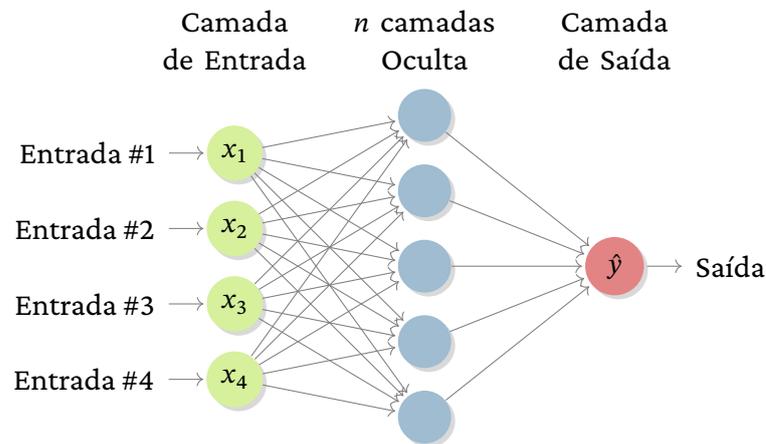
ReLU é considerada uma boa escolha para o otimizador (Figura 9) pois é muito similar a unidades lineares. A diferença entre ela e uma unidade linear é que ReLU resulta em 0 em metade do domínio. Esse fato faz com que a derivada por meio da unidade linear retificada seja grande independente se a unidade está ativa. E os gradientes não são grandes, mas também são consistentes. A segunda derivada da operação de retificação é 0 em quase toda parte, e a derivada dessa mesma operação é 1 em quase toda parte quando a unidade está ativa. Ou seja, a direção do gradiente é muito mais útil para o aprendizado do que o seria com funções que possuem efeitos de segunda ordem. Utilizando esta função, agora a camada é capaz de fazer a operação mostrada pela Equação (2.3) (GOODFELLOW; BENGIO; COURVILLE, 2016):

$$\hat{y} = \text{relu}(w \cdot x + b). \quad (2.3)$$

Após cada camada e suas unidades chegarem aos seus pesos ao final de uma iteração, o algoritmo de *back-propagation* entra em ação. Para tanto, faz-se necessária a utilização do otimizador, que determina como a rede será melhorada baseado no resultado da função de perda, implementando uma variante específica de SGD (do Inglês, *Stochastic Gradient Descent*, ou decréscimo estocástico do gradiente, em tradução livre). Além do SGD, outras escolhas tradicionais de otimizador são o Adam, Adamax e RMSprop (CHOLLET, 2017). A repetição desse processo conclui

o laço de treinamento ilustrado pela Figura 8. Um modelo esquemático de uma rede neural pode ser ilustrado pela Figura 10.

Figura 10 – Ilustração dos elementos típicos de uma rede neural. A camada de entrada está representada em verde, a camada oculta em roxo e a de saída em vermelho. Cada círculo representa um neurônio.



Fonte: Adaptado de Shiruru (2015)

2.2.1 Tipos de Aprendizado

Existem quatro grandes classificações no aprendizado de máquina, determinadas a partir de como os algoritmos aprendem. São eles: aprendizado supervisionado, não supervisionado, autossupervisionado e por reforço (CHOLLET, 2017; GOODFELLOW; BENGIO; COURVILLE, 2016).

O *aprendizado supervisionado* é aquele em que é entregue para o algoritmo um conjunto de exemplos de dados de entrada e seus respectivos resultados, e o aprendizado é realizado ao se estabelecer o mapeamento entre os dois. Em outras palavras, o aprendizado supervisionado envolve observar uma quantidade de exemplos de entrada x , devidamente associados aos respectivos y de saída, e construir uma generalização da relação entre os dois conjuntos (GOODFELLOW; BENGIO; COURVILLE, 2016). Esse tipo de aprendizado é mais comumente utilizado, possuindo aplicações em quase todas as áreas de destaque do *Deep Learning* nos dias atuais, como classificação de imagem, reconhecimento de fala e tradução de línguas. Mesmo sendo um aprendizado voltado para problemas de classificação ou regressão, existem variantes que podem ser utilizadas para geração de sequências, predição de árvores sintáticas, detecção de objetos e segmentação de imagens (CHOLLET, 2017).

No *aprendizado não supervisionado* as transformações pertinentes dos dados de entrada são aprendidas sem que exista um objetivo específico para direcionar o aprendizado. Esse tipo de aprendizado é utilizado nos conjuntos de dados que possuem muitas características, permitindo inferir quais são as propriedades realmente úteis na estrutura desse conjunto de dados. Geralmente é utilizado em áreas como visualização, compressão ou para eliminar ruídos de dados, e também é aplicado para encontrar correlações. É um passo importante para entender um conjunto de dados

antes de utilizar um algoritmo de aprendizado supervisionado (CHOLLET, 2017; GOODFELLOW; BENGIO; COURVILLE, 2016).

O *aprendizado autossupervisionado* começou como um caso particular do supervisionado, porém pode ser considerado diferente o suficiente para ser classificado separadamente. Nesse ramo os rótulos ou conjunto de saída y não são fornecidos explicitamente, e sim gerados a partir dos dados de entrada, geralmente com o uso de heurísticas. A estrutura de rede neural mais conhecida utilizada nessa forma de aprendizado é chamada codificador automático, (do Inglês, *autoencoder*) (CHOLLET, 2017).

As fronteiras entre esses três tipos clássicos de aprendizado podem não ser totalmente claras, pois essas categorias são mais próximas de algo contínuo, sem uma divisão rígida e inflexível. Dependendo da interpretação, o aprendizado autossupervisionado pode ser entendido como supervisionado ou não supervisionado, conforme o contexto da aplicação (CHOLLET, 2017).

Já o aprendizado por reforço é um ramo novo do aprendizado de máquina, e vem obtendo bastante atenção após um algoritmo do *DeepMind*¹ da Google conseguir aprender como jogar videogames do Atari 2600 (MNIH *et al.*, 2013), e em seguida a jogar Go, um jogo estratégico chinês, no mais alto nível (SILVER *et al.*, 2016; SCHRITTWIESER *et al.*, 2020). Esse tipo de aprendizado funciona recebendo informações sobre o problema, e aprendendo a escolher ações que maximizem algum tipo de recompensa. Por exemplo, uma rede neural que consiga jogar um videogame e busca maximizar sua pontuação no jogo pode ser treinada por meio do aprendizado por reforço. Essa ainda é uma área majoritariamente de pesquisa, sem muitos resultados práticos significativos além dos jogos. No entanto, espera-se que em algum momento futuro, essa forma de aprender domine as aplicações no mundo real, com aplicações em carros autônomos, robótica e gestão de recursos entre outras (CHOLLET, 2017)

2.2.2 Erros

Quando um modelo é avaliado, os dados que serão apresentados ao algoritmo são divididos em três conjuntos: treino, validação e teste. A função do conjunto de treino vem de seu nome: treinar o modelo. A qualidade do treino é avaliada com base nas generalizações feitas com os dados de validação. Finalmente, uma vez com o modelo pronto, a capacidade de generalização é avaliada nos dados de teste (CHOLLET, 2017).

O principal desafio do Aprendizado de Máquina é justamente a capacidade de generalização dos algoritmos: estes devem funcionar bem em amostras de dados inéditas, não apenas naquelas nas quais o modelo foi treinado. Existe uma tensão entre a otimização e a generalização: a otimização refere-se ao quão bem o modelo se ajusta aos dados de treino, ou seja, quando ele atinge o melhor desempenho para esse conjunto específico de dados; já a generalização trata de

¹ *DeepMind* é uma empresa britânica de inteligência artificial comprada pelo Google em 2014, com o objetivo de combinar as melhores técnicas de aprendizado de máquina e de sistemas de neurociência para construir algoritmos de aprendizado mais poderosos.

quão bem um modelo treinado age sobre dados que nunca antes foram vistos (GOODFELLOW; BENGIO; COURVILLE, 2016; CHOLLET, 2017).

Via de regra, durante o treinamento, o algoritmo tem acesso a um conjunto de dados usado exclusivamente para esse fim. É possível avaliar métricas de erro a partir desse conjunto — o chamado erro de treinamento — e tentar reduzi-lo, para assim avaliá-lo. No entanto, isso seria apenas uma otimização para esses dados específicos. O que difere o aprendizado de fato de uma simples otimização é justamente a busca por um erro generalizado, ou erro de teste, que também deve ser minimizado. O *erro generalizado* é definido pelo valor esperado do erro com uma nova entrada. Tipicamente ele é estimado a partir de um conjunto de exemplos de teste e de validação, que são coletados separadamente dos dados de treino (GOODFELLOW; BENGIO; COURVILLE, 2016; CHOLLET, 2017).

Os dados são separados em três, pois o processo de desenvolvimento de um modelo envolve o ajuste de suas configurações (chamadas de hiperparâmetros), como número e tipos de camadas e número de neurônios. Esses ajustes são feitos usando o desempenho do modelo nos dados de validação. Na sua essência, esses ajustes são também uma forma de aprendizado, pois são uma busca por uma configuração otimizada em algum espaço de parâmetros (CHOLLET, 2017). Durante esse processo, o valor esperado do erro de validação é maior ou igual ao valor esperado do erro de treino. Logo os fatores que determinam quão bem o algoritmo vai performar são (GOODFELLOW; BENGIO; COURVILLE, 2016):

- Tornar erro de treinamento o menor possível;
- Minimizar sua distância em relação ao erro de validação.

Esses dois fatores correspondem a dois problemas centrais no Aprendizado de Máquina, o *underfitting*, e *overfitting* (em tradução livre do Inglês, respectivamente, *subajuste* e *sobreajuste*). O *underfitting* é associado ao primeiro fator, ou seja, quando o modelo não tem a capacidade de obter um erro baixo o suficiente no conjunto de treino, logo não se ajustando bem aos dados. O *overfitting* ocorre quando a distância entre os dois erros é muito grande (GOODFELLOW; BENGIO; COURVILLE, 2016).

Quando se inicia o treinamento de uma rede, a otimização e a generalização estão relacionadas, ou seja, as diminuições nos valores da função de perda nos dados de treino e nos dados de validação ocorrem de maneira associada. Após um certo número de iterações, a generalização para de melhorar, ao mesmo tempo que as métricas de validação param de evoluir e começam a se degradar, caracterizando o fenômeno de *overfitting*: a rede está aprendendo padrões que são específicos dos dados de treino, mas são irrelevantes ou errados para dados inéditos (CHOLLET, 2017).

Pode-se controlar se um modelo terá uma tendência ao *overfitting* ou ao *underfitting* alterando sua capacidade. A capacidade de um modelo é sua habilidade de se ajustar a uma ampla

gama de funções, ou seja, um modelo com baixa capacidade pode ter dificuldade de se ajustar ao conjunto de treino. Já um modelo com alta capacidade pode sofrer de *overfitting*, por memorizar as propriedades de treino que não funcionam bem no conjunto de testes (GOODFELLOW; BENGIO; COURVILLE, 2016).

Uma maneira de controlar a capacidade de um modelo é a partir da escolha de seu espaço de hipóteses. Para exemplificar, um caso de regressão linear pode ser generalizada para incluir diversos polinômios, além das funções lineares no seu espaço de hipótese, o que aumenta a sua capacidade (GOODFELLOW; BENGIO; COURVILLE, 2016).

Um polinômio de grau 1 nos dá um modelo de regressão linear com uma predição de acordo com a Equação (2.4)

$$\hat{y} = b + wx. \quad (2.4)$$

Aumentando um grau no polinômio, pode-se aprender usando um modelo quadrático, como o da Equação (2.5)

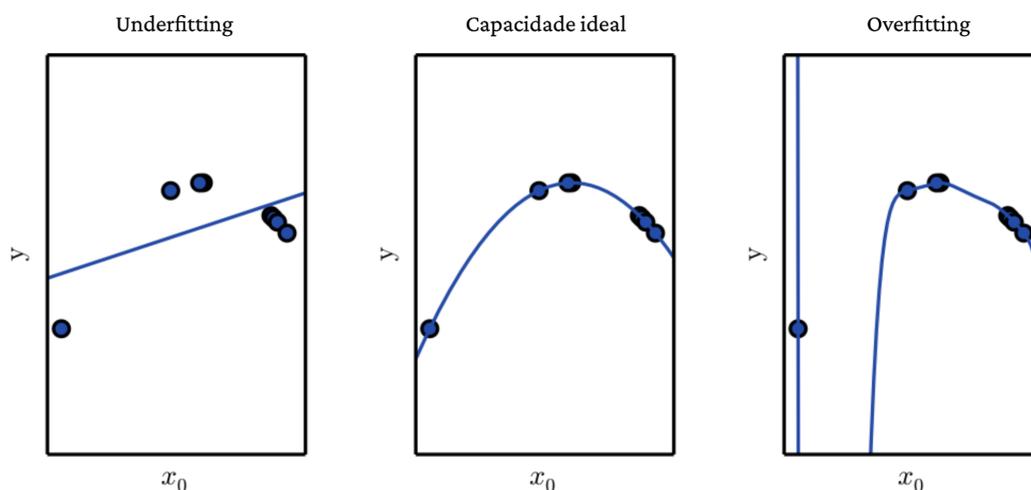
$$\hat{y} = b + w_1x + w_2x^2. \quad (2.5)$$

Esse processo de aumento de capacidade pela adição de potências maiores de x pode, por exemplo, levar a um polinômio de grau 9. No entanto, os algoritmos de Aprendizado de Máquina vão realizar seu melhor desempenho quando sua capacidade é apropriada para a complexidade da tarefa e para a quantidade de dados que lhes é oferecida. Modelos com baixa capacidade não são capazes de resolver problemas complexos, mas modelos com altas capacidades podem apresentar *overfitting* em tarefas mais simples, mesmo sendo capazes de resolver problemas complexos (GOODFELLOW; BENGIO; COURVILLE, 2016).

A Figura 11 ilustra esse princípio. Quando se compara uma predição com um modelo linear, um quadrático e um de grau 9, todos tentando resolver um problema quadrático, nota-se a diferença dos resultados obtidos com as diferentes capacidades. A função linear não é capaz de realizar a curvatura entre os pontos, por isso não se adapta. A função de grau 9 é capaz de representar todos os pontos, no entanto tem-se mais parâmetros do que exemplos de treinamento. Existe uma dificuldade em escolher uma solução que se generalize bem quando existem tantas opções possíveis. Nesse exemplo, o modelo quadrático é o que se encaixa perfeitamente à estrutura real do problema, então generaliza bem aos novos dados (GOODFELLOW; BENGIO; COURVILLE, 2016).

Quando se otimiza uma rede, no entanto, escolher a configuração do modelo apenas baseada nos dados de desempenho a partir dados de validação pode rapidamente resultar em um *overfitting* do conjunto de validação, mesmo que o modelo nunca tenha aprendido a partir dele de maneira direta (CHOLLET, 2017). O desempenho de um modelo a partir dos dados de validação sempre atinge um pico após uma quantidade de iterações, começando a piorar na sequência. Em outras palavras, a partir de certo ponto, a rede começa a apresentar o *overfitting*, que pode ocorrer em todo problema de aprendizado (CHOLLET, 2017).

Figura 11 – Ilustração do que é um *underfitting* (à esquerda), um bom ajuste (ao centro), e um *overfitting* (à direita).



Fonte: Goodfellow, Bengio e Courville (2016, p. 110).

Para prevenir o aprendizado de padrões que levem ao *overfitting*, a melhor solução é obter mais dados de treino. Um modelo treinado com mais dados naturalmente apresenta uma maior generalização. Quando isso não é possível, a segunda melhor opção torna-se modular a quantidade de informação que aquele modelo é capaz de guardar, ou adicionar restrições em quais informações são permitidas. Se a rede só for capaz de memorizar um pequeno número de padrões significativos, existe uma maior chance de se obter uma boa generalização (CHOLLET, 2017).

O processo de tentar evitar o *overfitting* é chamado de *regularização*. Algumas técnicas comuns de regularização são a redução do tamanho da rede, a adição de pesos de regularização, e o uso de *dropout* (em tradução livre do Inglês, *deixado de fora*) (CHOLLET, 2017).

Reduzir o tamanho da rede é o jeito mais simples de reduzir a probabilidade de *overfitting*, pois reduz o número de parâmetros aprendidos no modelo, reduzindo sua capacidade (CHOLLET, 2017). No entanto, nesse caso é importante atentar-se para um conceito: o vazamento de informação. Toda vez que um hiperparâmetro de um modelo é ajustado baseado no desempenho nos dados de validação, algumas informações sobre os dados de validação “vazam” para dentro do modelo. Se isso é feito uma vez, para apenas um parâmetro, seu conjunto de validação continuará sendo confiável. No entanto, se for repetido diversas vezes, pode aumentar significativamente a quantidade de informação vazada para dentro do modelo, o que resulta em uma performance artificialmente boa nos dados de validação, pois está otimizado para tal (CHOLLET, 2017).

Adicionar pesos de regularização é outra forma de tornar os modelos mais simples, a partir da adição de multiplicadores nos pesos, diminuindo a entropia da distribuição de parâmetros, o que gera uma menor chance de *overfitting*, pois os pesos da rede são forçados a serem apenas valores pequenos, gerando uma distribuição mais regular. No entanto, isso não acontece sem um custo. As mais comuns são as regularizações L_1 que adiciona um custo proporcional ao módulo dos coeficientes dos pesos, e L_2 , que adiciona um custo proporcional ao quadrado dos coeficiente

dos pesos. Também é possível combinar regularizações (CHOLLET, 2017).

E por último tem-se o *dropout*, que é uma das técnicas mais comuns e efetivas para regularizar modelos. O *dropout* consiste em aleatoriamente transformar em zero alguns dos números de saída de cada camada durante o treino. Exemplificando, se uma camada, por exemplo, retornaria um vetor $[0,2; 0,5; 1,3; 0,8; 1,1]$, para uma determinada entrada, após aplicar o *dropout*, alguns desses valores se tornarão zero, escolhidos aleatoriamente, como por exemplo, $[0; 0,5; 1,3; 0; 1,1]$. A taxa de *dropout* é escolhida a priori e representa qual fração das características será zerada, geralmente ficando entre 0,2 e 0,5 (CHOLLET, 2017).

2.3 Metamodelos

Como dito por Pietrocola (2002), a Matemática é a linguagem estruturante do pensamento físico. Muitos fenômenos físicos podem ser representados por meio de modelos matemáticos na forma $y = f(x)$ (FANG; LI; SUDJIANTO, 2005). De acordo com Greca e Moreira (2002), o trabalho de um cientista está no processo, tanto na geração ou na aplicação de um modelo físico, de trazer junto um processo semântico, no qual será traduzido o modelo do fenômeno para a linguagem matemática, para assim ter-se uma descrição completa de acordo com os cânones científicos.

Nesse tipo de representação, x é um vetor de variáveis de entrada, definido nos reais, e y é uma variável de saída escalar, que representa a resposta de $f(x)$ que está sendo modelada. No entanto, há situações em que calcular $f(x)$ pode ser computacionalmente custoso ou levar um tempo proibitivo. Neste contexto, existem metodologias de simulação de modelos matemáticos que, por meio de aproximações numéricas permitem explorar o modelo e conseqüentemente o fenômeno. Pode-se citar como exemplos o método de Elementos Finitos e o CFD (do Inglês, *Computational Fluid Dynamics, Dinâmica de Fluidos Computacional*). Essas aproximações são feitas encontrando um \hat{y} que seja próxima o suficiente de $f(x)$; esse processo recebe o nome de *metamodelo*, que é uma resposta acurada, no entanto mais barata e mais rápida de se atingir.

O processo de construir modelos aproximados, também chamado de *metamodelagem*, é iterativo, por definição. Cinco passos devem ser seguidos:

- Definição de um espaço de projetos;
- Realização de um projeto de experimento (DOE, do Inglês, *design of experiment*);
- Avaliação do modelo;
- Criação de metamodelos;
- Validação dos metamodelos.

Faz sentido falar sobre metamodelagem e aprendizado de máquina de maneira conjunta, pois a primeira nada mais é que uma técnica de aprendizado supervisionado, onde os dados de

um modelo original mais custoso são utilizados para treinar um algoritmo de aprendizado, capaz de reproduzir com alguma fidelidade o modelo original de maneira mais barata, seja em tempo computacional ou em custos associados ao processo de obtenção dos dados. Logo, a relação entre criar um metamodelo e treinar um algoritmo de aprendizado de máquina é direta.

Com um metamodelo já validado, muitas atividades de engenharia como estudos preliminares e visualizações gráficas, predição e otimização, para citar algumas, podem ser realizadas mais rapidamente.

De maneira geral, a maior parte dos metamodelos podem ser criados utilizando-se de uma combinação linear de funções de base $B = \{B_1(x), B_2(x), B_3(x), \dots, B_L(x)\}$, com $B_k : \mathbb{R}^n \rightarrow \mathbb{R}$; logo, um metamodelo $\hat{y}(x)$ pode ser escrito:

$$\hat{y}(x) = \sum_i \beta_i B_i(x), \quad (2.6)$$

onde β_i , são coeficientes a serem encontrados.

Existem diferentes métodos para escolha das bases, cada um deles trazendo diferentes propriedades, como simplicidade conceitual, transparência, acurácia, robustez, eficiência, e custo computacional. Explicações mais abrangentes acerca desses métodos podem ser encontradas nas referências Fang, Li e Sudjianto (2005) e Sobester, Forrester e Keane (2008).

Na análise e otimização de engenharia, os métodos mais comuns são:

- Resposta de superfície polinomial;
- *Kriging*;
- Funções de base radiais;
- Redes Neurais (foco deste trabalho);
- Regressão de vetores de suporte.

Em resumo, para construir um metamodelo, leva-se em conta a relação funcional entre a variável de entrada x e a de saída y , dada por (2.7):

$$y = f(x), \quad (2.7)$$

e a partir dela, cria-se um metamodelo na forma da Equação (2.6), dado por (2.8). Ou seja, se substitui a função objetivo real, custosa e complexa, e todas suas restrições, por seus respectivos mais baratos e rápidos.

$$\hat{y} = g(x) = \sum_i \beta_i B_i(x), \quad \text{sendo } y = \hat{y} + \epsilon, \quad (2.8)$$

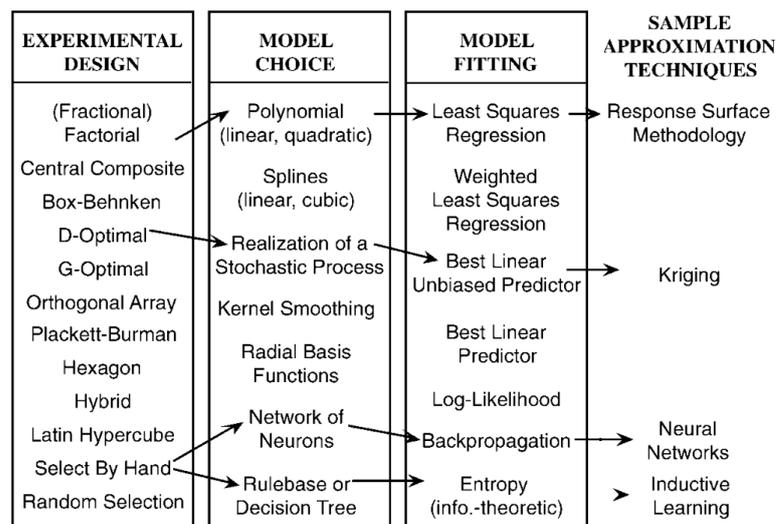
onde \hat{y} é uma aproximação de y e ϵ é o erro da aproximação e dos erros de medição (FERREIRA, 2016; SIMPSON, T. W. *et al.*, 2001).

Atualmente sabe-se que não existe um método de aproximação universal para propósitos de metamodelagem. A melhor estratégia é fortemente dependente de fatores como o grau de não-linearidade do problema, o número de variáveis de design, a técnica de amostragem, a forma funcional do modelo substituto (parâmetros de forma e ajuste) e a medida de erro adotada.

De maneira geral, a metamodelagem é um processo usado na criação de uma abstração, seja ela uma aproximação ou uma interpolação, de um determinado fenômeno em um certo domínio, criando um “modelo do modelo”. A resposta desse metamodelo será avaliada pela comparação com um experimento físico, ou com uma simulação computacional bem estabelecida com um método de alta fidelidade em alguns pontos do domínio (SIMPSON, T.W. *et al.*, 2008).

Os cinco passos supra-citados, podem ser unidos em três pontos importantes destacados por T. W. Simpson *et al.* (2001): a escolha de um projeto de experimento, ou, do Inglês, *design of experiment* (DOE) para gerar os dados, a escolha de um modelo para representá-los e uma maneira de ajustar o modelo a esses. Existem inúmeras combinações possíveis para esses três passos, como pode ser visto na Figura 12.

Figura 12 – Diferentes técnicas de metamodelagem.



Fonte: T. W. Simpson *et al.* (2001).

O que é comum a todas essas diferentes técnicas é o fato de que a resposta “real” do material — seja ela obtida através de experimentos em laboratório, ou através de modelos computacionais de alta precisão — é conhecida em um conjunto Ω finito e predeterminado de pontos. O metamodelo é criado para um subconjunto de Ω , e pode, em seguida, substituir o modelo computacional que seria utilizado originalmente (SIMPSON, T.W. *et al.*, 2008).

A otimização baseada em metamodelos mais comum funciona de maneira que um DOE é criado e um modelo de simulação é usado para avaliar as performances $f_i(x)$. Este processo é repetido até que sejam encontradas aproximações $\hat{f}_i(x)$ boas o suficiente dentro de uma métrica pré-definida. Ao final, o algoritmo de otimização usa os metamodelos $\hat{f}_i(x)$ válidos para buscar o

projeto ótimo (FERREIRA, 2016; SIMPSON, T. W. *et al.*, 2001).

Uma das principais vantagens de se realizar a metamodelagem é que essas podem substituir os códigos de análise que já existem fornecendo ótimos resultados, como enumera T. W. Simpson *et al.* (2001):

1. Um melhor entendimento da relação entre x e y ;
2. Uma integração mais fácil de códigos de computador dependentes de domínio;
3. Ferramentas de análise rápida para otimização e exploração do espaço de design utilizando-se de aproximações no lugar dos códigos de análise que são computacionalmente mais caros.

Essa metodologia já mostrou bons resultados em cenários de otimização multidisciplinar e multiobjetivo, e vem sendo muito utilizada tanto na indústria quanto na pesquisa acadêmica, como por exemplo nas áreas automotiva e aeroespacial (FERREIRA, 2016).

3 Compósitos Poliméricos Reforçados por Fibra de Carbono

Entre os diferentes compósitos citados anteriormente, aqueles reforçados por fibra de carbono, em Inglês, *carbon fiber-reinforced polymers* (CFRP), são um tipo que merece o destaque, sendo portanto, foco deste trabalho. Desde a sua descoberta em 1964, pela Royal Aircraft do Reino Unido, a fibra de carbono vem sendo cada vez mais utilizada em materiais compósitos (LORANDI; CIOFFI; JUNIOR, 2016; MOURITZ, 2012).

As fibras de carbono são muito rígidas, robustas, e têm filamentos leves, por isso são usadas em compósitos de matriz polimérica, geralmente epóxi. Desde a década de 1990 esses compósitos estão sendo amplamente utilizados na indústria aeroespacial, sendo que hoje, segundo Mouritz (2012), estão lado a lado com o alumínio como material mais utilizado na parte estrutural.

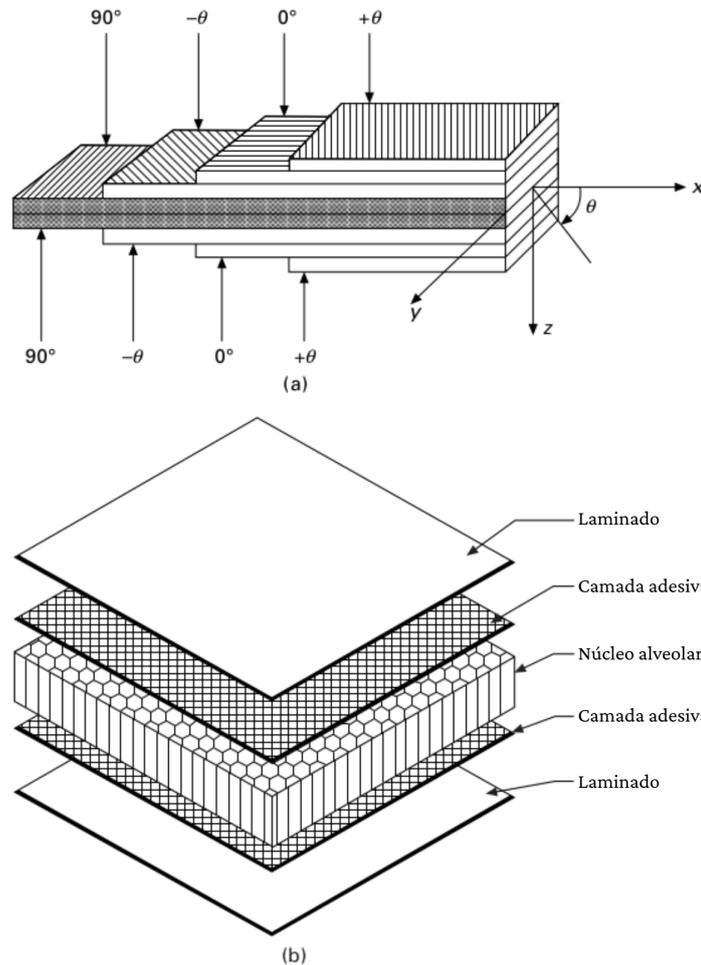
As matrizes poliméricas, tanto as matrizes termorrígidas como as termoplásticas, embora possuam baixa massa específica, são menos resistentes e rígidas que as fibras. Isso faz com que as propriedades mecânicas dos compósitos poliméricos sejam influenciadas diretamente tanto pela orientação das fibras que o reforçam em relação às solicitações mecânicas, como pela fração volumétrica dos constituintes iniciais. Essa dependência também existe nas propriedades higrótérmicas dos compósitos poliméricos, como condução de calor, dilatação térmica e absorção de umidade, por exemplo (NETO; PARDINI, 2016).

O processo de manufatura deste material implica diretamente em seu projeto final e também implica em quais propriedades físicas ele terá, sendo afetado pela fração de volume e a orientação das fibras de reforço. Os compósitos desse tipo utilizados em aeronaves são manufaturados em duas formas básicas: estruturas laminadas e sanduíches (MOURITZ, 2012).

A estrutura laminada consiste em várias camadas de fibra e resina, chamadas de folhas, que são unidas em um material sólido. Nesse caso as fibras são orientadas ao longo das direções no plano de carga principal, de maneira a obter uma alta dureza, robustez, e resistência à fadiga, sendo a matriz polimérica o que junta as fibras ao material. A estrutura sanduíche consiste em duas finas superfícies, usualmente carbono-epóxi laminados, ligados por um filme adesivo a um material principal espesso e leve (MOURITZ, 2012). As duas estruturas podem ser vistas na Figura 13. O foco deste trabalho está nas estruturas laminadas.

As propriedades mecânicas dos compósitos podem ser projetadas sob medida devido aos alinhamentos dos reforços de fibra, o que faz com que se tenha maior rigidez e maior força onde necessário. Assim, os valores específicos das propriedades de rigidez e força dos compósitos de fibra de carbono são superiores aos da liga de alumínio, garantindo ainda maior flexibilidade no projeto de componentes estruturalmente eficientes (MOURITZ, 2012).

Figura 13 – Duas formas de construção de um compósito: em (a) tem-se uma estrutura laminada, composta de lâminas com fibras em ângulos variados e em (b) tem-se a estrutura em sanduíche.



Fonte: Mouritz (2012, p. 305).

3.1 Propriedades Mecânicas

Quando uma estrutura mecânica sofre a ação de forças externas, sejam elas forças de superfície — como por exemplo a torção de uma barra — ou forças de corpo — como por exemplo, o peso de poste iluminando a rua sobre si mesmo — são geradas forças internas. O conhecimento dessas forças internas em todos os pontos do corpo é primordial, pois elas precisam ser menores que a resistência do material utilizado na estrutura. A tensão é definida a partir desse fenômeno, sendo a intensidade de carga por unidade de área.

Em um caso hipotético de um corpo em equilíbrio sob a ação de diversas cargas, um corte transversal nesse corpo fará com que as forças sejam aplicadas na área da seção transversal para manter o equilíbrio, como no corpo original. Em qualquer ponto desta seção, existe uma força ΔP atuando na área ΔA . Logo, o vetor de força tem uma componente normal, ΔP_n , e uma paralela à

superfície, ΔP_s . A definição de tensão então é dada por:

$$\sigma_n = \lim_{\Delta A \rightarrow 0} \frac{\Delta P_n}{\Delta A}, \quad (3.1)$$

$$\tau_s = \lim_{\Delta A \rightarrow 0} \frac{\Delta P_s}{\Delta A} \quad (3.2)$$

onde a tensão normal é dado pela Equação (3.1) e a tensão de cisalhamento dado pela Equação (3.2). A mesma definição se aplica quando se trata de um corpo em três dimensões, resultando em uma tensão normal numa direção, e dois de cisalhamento, nas duas outras direções, para cada face, logo existindo para cada ponto, 6 diferentes tensões (KAW, 2007)

De maneira similar ao conhecimento das forças dentro de um corpo, conhecer as alterações de forma por conta das forças externas também é importante. Por exemplo, um pistão em um motor de combustão interna pode atuar sem desenvolver uma maior tensão que as resistências de falha, no entanto uma alteração de forma excessiva pode travar o motor. Para encontrar a tensão em um corpo precisa-se encontrar essas alterações, porém a tensão em um ponto tem seis componentes, mas apenas três equações de força-equilíbrio, uma em cada direção.

O conhecimento das alterações de forma é especificado em termos de *deformações*, ou seja, a mudança relativa no tamanho e forma do corpo. Essa deformação em um ponto é geralmente definida em um paralelepípedo infinitesimal em um sistema de coordenadas locais. Sob o efeito de cargas, as dimensões dos lados desse paralelepípedo infinitesimal mudam. As faces do cubo também são distorcidas. As mudanças nas dimensões correspondem à deformação normal e a distorção corresponde à deformação de cisalhamento.

De maneira geral, para corpos elásticos, o módulo elástico ou *módulo de Young* em uma direção, pode ser calculado como a razão entre a tensão e a deformação desse corpo nessa determinada direção, representado pela Equação (3.3)

$$E_i \equiv \frac{\sigma_i}{\epsilon_i}, \quad (3.3)$$

onde σ_i é a tensão e ϵ_i é a deformação, ambos em uma direção i (KAW, 2007).

Um dos principais motivos para se utilizar compósitos de fibras contínuas, principalmente em aeronaves, é devido à sua alta rigidez específica em comparação com muitas ligas metálicas. O módulo de Young dos CRFP são muito superiores ao do alumínio, por exemplo. O módulo de Young de um compósito reforçado por fibras retas e contínuas quando sofre uma carga na direção das suas fibras pode ser calculado usando a modelagem da lei das misturas (do Inglês *rule-of-mixtures modelling*). Em um compósito unidirecional, sob uma tração na direção das fibras, assume-se que suas fibras e sua matriz estão agindo em paralelo e ambos os constituintes sofrem de uma mesma deformação elástica (ϵ_1). Essa condição é verdade mesmo que os módulos das fibras e da matriz sejam diferentes, e é expressa por:

$$\epsilon_1 = \epsilon_f = \frac{\sigma_f}{E_f} = \epsilon_m = \frac{\sigma_m}{E_m}, \quad (3.4)$$

onde σ_f e σ_m são a tensão na fibra e na matriz, e E_f e E_m o módulo de Young de cada uma delas.

Assim, pode-se calcular o módulo de Young longitudinal por meio da Equação (3.5)

$$E_1 = E_f V_f + E_m V_m, \quad (3.5)$$

onde V_f e V_m são a porcentagem de fibra e matriz; se o material está inteiro $V_f + V_m = 1$. A lei das misturas, no entanto só é precisa para volumes de fibra entre 0,2 e 0,7.

Já que o módulo elástico da matriz é muito menor que o das fibras, geralmente essas contribuem cerca de 95 % a 99 % da rigidez no plano em um compósito unidirecional com fibras de carbono com um $V_f \approx 0,55 - 0,65$, costumeiramente utilizado na indústria aeroespacial. Já o tipo da matriz polimérica não tem muito efeito, sendo escolhida geralmente por outros motivos como custo, temperatura máxima de operação ou durabilidade.

Compósitos são geralmente projetados para garantir que a carga seja aplicada sempre paralelamente às fibras, por conta desses possuírem baixa rigidez e resistência transversal. Para calcular o módulo elástico transversal assume-se que as fibras e a matriz atuam em série em relação à carga externa, fato expresso pela Equação (3.6)

$$E_2 = \left[\frac{V_f}{E_f} + \frac{V_m}{E_m} \right]^{-1}. \quad (3.6)$$

Os reforços têm muito menos efeito no módulo de Young transversal do que no longitudinal. A Equação (3.6) não é sempre exata para calcular o módulo transversal, por conta da deformação não ser uniforme quando um compósito sofre carga na direção oposta das fibras.

Existem outras maneiras de obter o módulo de Young transversal como o modelo de Halpin-Tsai, que leva em conta a distribuição heterogênea pela matriz polimérica, dada pela Equação (3.7)

$$E_2 = \frac{E_m(1 + \xi \nu V_f)}{(1 - \nu V_f)}, \quad (3.7)$$

onde ν é a taxa de interação, dada por:

$$\nu = \frac{\frac{E_{2f}}{E_m} - 1}{\frac{E_{2f}}{E_m} + \xi}, \quad (3.8)$$

sendo ξ um parâmetro ajustável (MOURITZ, 2012).

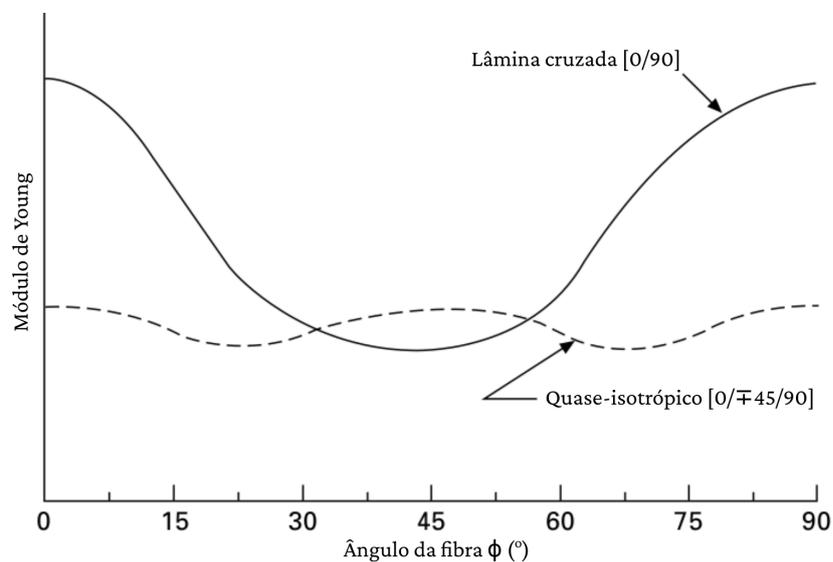
O módulo de Young de um compósito unidirecional é muito maior quando a carga é na direção longitudinal que na direção transversal devido à alta dureza proporcionada pelas fibras. Em casos de compósitos de carbono-epóxi de alto módulo, essa diferença chega a ser de 200 vezes entre as grandezas. O módulo de Young de um compósito unidirecional sofrendo a carga com a fibra em qualquer ângulo entre 0° e 90° pode ser calculado pela Equação (3.9)

$$E(\Phi) = \left[\frac{\cos^4 \Phi}{E_1} + \frac{\sin^4 \Phi}{E_2} + \left(\frac{1}{G_{12}} - \frac{2\nu_{12}}{E_1} \right) \sin^2 \Phi \cos^2 \Phi \right]^{-1}, \quad (3.9)$$

onde G_{12} é o módulo de cisalhamento no plano, e ν_{12} é a razão de Poisson no plano (MOURITZ, 2012)

Por conta dessa natureza altamente ortotrópica das propriedades de um compósito unidirecional, ele raramente é usado dessa maneira como material estrutural. Geralmente são usados, em aeronaves por exemplo, aqueles com fibras alinhadas em duas ou mais dimensões. As mais comuns são as lâminas cruzadas (do Inglês, *cross-ply*) que são $[0/90]$, ou seja 50 % das fibras alinhadas em cada direção, e o $[0/\pm 45/90]$ que são chamados de quase-isotrópicos e possuem 25 % das fibras alinhadas em cada direção. O efeito dessas duas configurações em diferentes ângulos de deformação pode ser visto na Figura 14 (MOURITZ, 2012).

Figura 14 – Efeito no módulo de Young para cargas em diferentes ângulos para compósitos de dobra cruzada (*cross-ply*) e quase-isotrópicos.



Fonte: Adaptada de Mouritz (2012, p. 363).

4 Metodologia

Esta seção é dedicada à descrição dos procedimentos executados para o cumprimento do objetivo originalmente proposto para este trabalho.

Para tanto, inicia-se com a geração de dados para seu treinamento, utilizando o *Multiscale Designer* (ALTAIR ENGINEERING INC., 2021), com uma licença gentilmente cedida pela Universidade Federal do ABC. Esse software é utilizado para o desenvolvimento de modelos computacionalmente eficientes e precisos de materiais diversos submetidos a condições que podem variar desde comportamentos isotrópicos homogêneos a anisotrópicos heterogêneos. O *Multiscale Designer* é particularmente eficiente na simulação de compósitos.

4.1 Configuração inicial

Na tela inicial do software, mostrada na Figura 15, pode-se escolher o tipo de modelagem de material, sendo o escolhido o multiescala (aba *Material Model Development*, opção *Material Model Type*), indicado por (A). Ainda na Figura 15, também é possível escolher entre os desenvolvimentos determinísticos e estocásticos, sendo escolhido o determinístico (quadro *Deterministic*), como indicado em (B). Nelas serão posteriormente determinadas o modelo de célula unitária (*Unit Cell Model Definition*, Seção 4.2), a caracterização linear do material (*Linear Material Characterization*, Seção 4.3) e a não linear (*Nonlinear Material Characterization*, Seção 4.4). Cada botão será habilitado quando o anterior houver sido completado. É possível escolher um nome significativo para o material na opção *Material Model Name*, na seção indicada por (C). As escolhas no desenvolvimento do material compósito e suas simulações foram baseadas em Wollschlager (2020).

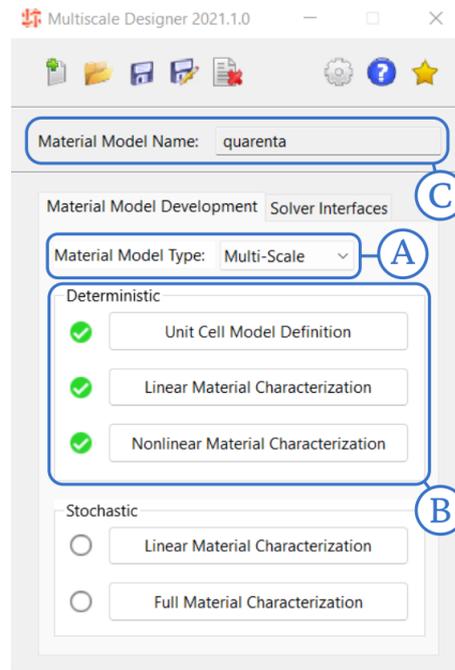
4.2 Modelo de célula unitária (RVE)

Dentro do desenvolvimento determinístico, inicia-se a simulação pela definição do modelo de célula unitária, ou RVE, exibido na Figura 16.

Nessa tela é possível escolher o tipo de seu modelo (*Model*), sua configuração (*Configuration*) e a maneira que serão fornecidos os dados de entrada (*Input Data*). Nesse trabalho utilizou-se o modelo fibroso (*Fibrous*), com a configuração em colmeia sem interface (*Honeycomb*), e com dados de entrada geométricos (*Geometric data*). Todas essas opções estão marcadas na Figura 16 com (A).

A escolha por dados de entrada geométricos determina que o modelo unitário será desenvolvido baseado na fração do volume de fibra. No exemplo da Figura 16 apresenta-se a construção de um modelo com 40 % de fibra (marcado com (B), em *Max Fiber Volume Fraction (%)*). Logo

Figura 15 – Tela inicial do *Multiscale Designer*, da Altair. Nela são exibidas o tipo de modelagem de material (aba *Material Model Development*, opção *Material Model Type*, indicado por (A)), o desenvolvimento determinístico (quadro *Deterministic*, indicado por (B)) e o nome escolhido para o material (indicado por (C)).



Fonte: Criada pelo autor.

abaixo dos parâmetros, em (C), é possível controlar como a malha será gerada para a simulação em Elementos Finitos (CHEN, 2005; HUGHES, 2012). Em todos os dados utilizados nesse trabalho a malha foi criada com elementos do tipo tetraédrico quadrático (opção *Element Type* em *Quad. Tetrahedron*), de maneira direta (*Direct*) e de tamanho de referência de 0,06 (*Element Size*).

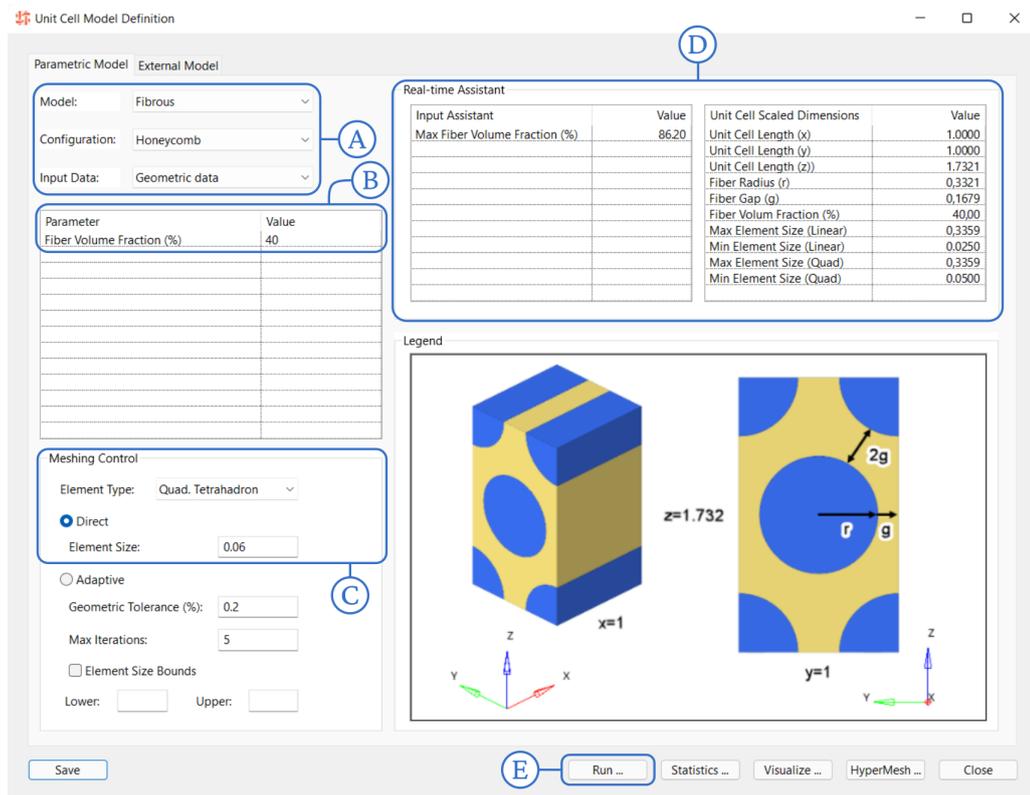
No canto superior esquerdo dessa tela, tem-se o assistente em tempo real (*Real-time Assistant*), que mostra o máximo de volume de fibra aceito pelo software (*Max Fiber Volume Fraction (%)*), e as dimensões escaladas da célula unitária (*Unit Cell Scaled Dimensions*), esses sendo gerados automaticamente a partir dos parâmetros já definidos. O RVE configurado de acordo com esses parâmetros é gerado clicando-se no botão executar (*Run...*), em (E). A malha do RVE pode ser apreciada na Figura 17, compreendo 36214 elementos tetraédricos de segunda ordem e 53883 nós.

4.3 Caracterização linear do material

Voltando à tela da Figura 15, com o modelo unitário pronto, aparecerá o sinal verde ✓, o que habilita o segundo botão, permitindo o acesso à janela de caracterização linear do material. Nessa tela, mostrada na Figura 18, é possível determinar os materiais que constituem esse compósito.

No destaque (A) (*Micro Material Property*) é possível selecionar separadamente os materiais para as duas fases do compósito, matriz e fibra (abas *Phase 1* e *Phase 2*), a partir do botão marcado

Figura 16 – Tela de definição do modelo de célula unitária do *Multiscale Designer*, da Altair. Em ① foi escolhido o modelo fibroso (*Model: Fibrous*), com configuração em colmeia (*Configuration: Honeycomb*) e dados de entrada em forma geométrica (*Input Data: Geometric data*). Em ② escolheu-se um material com 40 % de fibra (*Max Fiber Volume Fraction (%)*). A malha de Elementos Finitos definida em ③ é composta de elementos tetraédricos quadráticos (*Element Type: Quad. Tetrahedron*) com um tamanho de referência de 0,06 (*Element size*).



Fonte: Criada pelo autor.

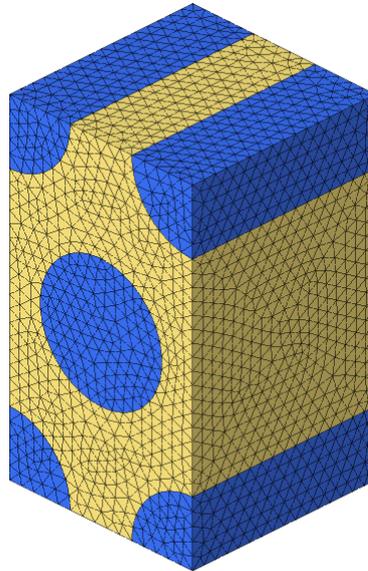
com ②. Os materiais são selecionados de uma base de dados interna do *Multiscale Designer* e foram um polímero termofixo de nome epóxi com módulo padrão (*Epoxy - Std Modulus - Continuous Products*) para a matriz com fibras de carbono de módulo padrão (*Carbon - Std Modulus - Continuous Fiber*), com ambos materiais na temperatura de 22 °C e com as unidades no sistema internacional SI.

Logo abaixo, marcado com ③, pode-se definir estruturas laminadas para simulações (*Laminate*). Foi definido que o compósito poderá ser formado por 4 diferentes estruturas (*Layups*), cada uma delas correspondendo a uma aba (*Layup 1* a *Layup 4*). Note que cada uma das 4 estruturas definidas de acordo com a Tabela 1 resulta em um compósito diferente. As repetições das estruturas 4 e 5 (*Layup 4* e *Layup 5*) ocorrem devido a presença da simetria, ativada pela opção *Symmetric*.

Os dados apresentados em ④ são provenientes da simulação linear, após terminada sua execução.

Uma representação meramente ilustrativa de como as fibras se organizam na estrutura

Figura 17 – Malha de um RVE com 40 % de fibras gerado no *Multiscale Designer*, da Altair. Ela é composta por 36214 elementos tetraédricos de segunda ordem e 53883 nós. A fibra está representada em azul e a matriz em amarelo.



Fonte: Criada pelo autor.

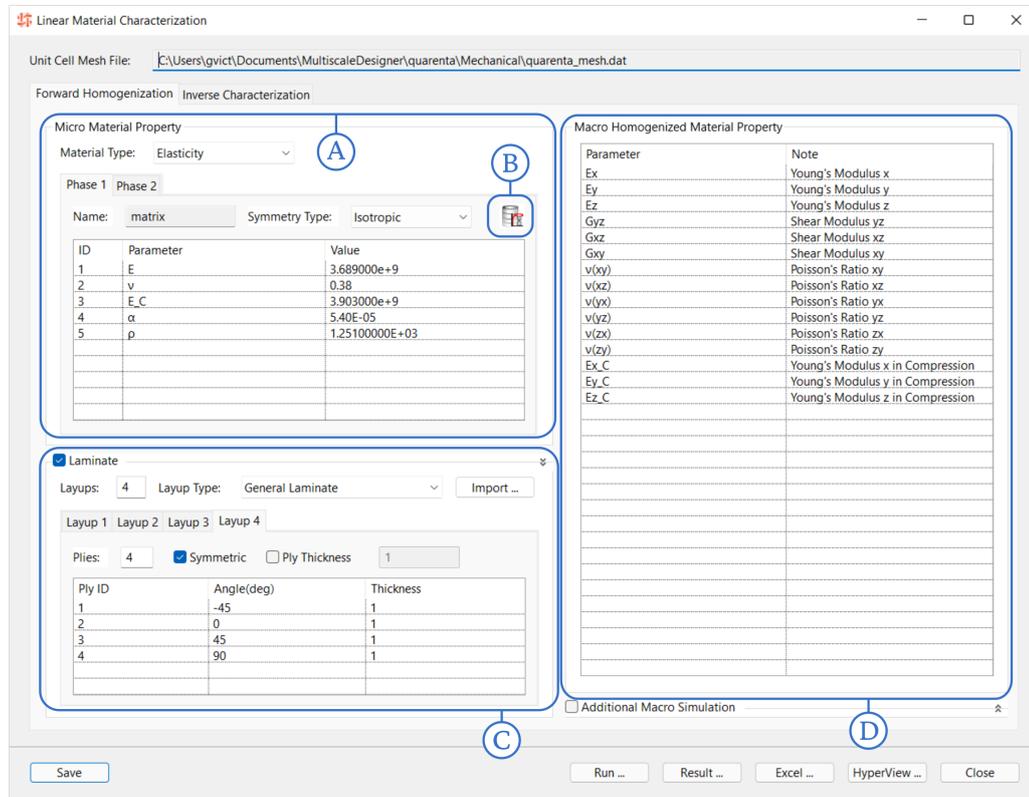
Tabela 1 – Configurações de matriz e fibra simuladas no *Multiscale Designer*. A configuração 4 também recebe o nome de *quase-isotrópica*. As escolhas das estruturas foram baseadas em Wollschlager (2020).

Estrutura	Folha	Ângulo (°)	Repetições	Total de folhas
1	1	0	1	1
2	1	90	1	1
3	1	45	2	4
	2	-45		
	1	-45		
4	2	0	2	8
	3	45		
	4	90		

Fonte: Criada pelo autor.

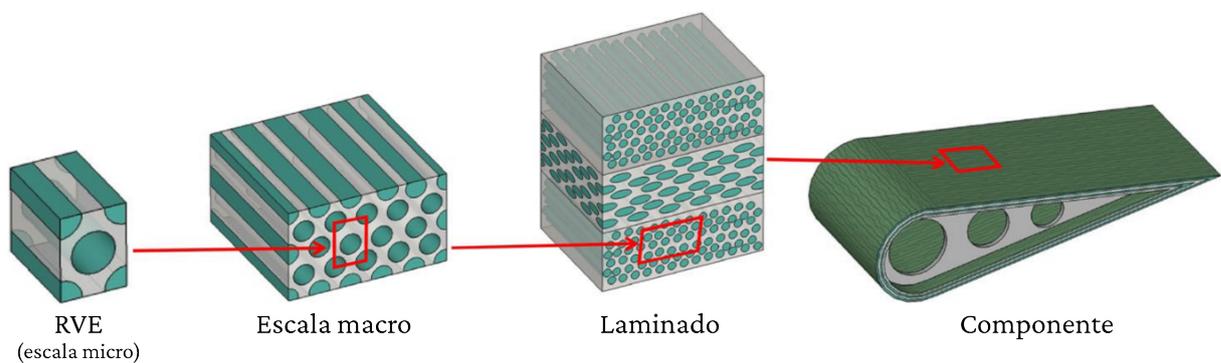
laminada é a Figura 19, que mostra uma escala crescente de tamanhos da esquerda para a direita. Partindo do RVE chega-se à escala macro, onde vários RVEs se combinam para formar a lâmina. O material laminado é construído com o empilhamento de lâminas com dois ângulos diferentes de fibra, que finalmente constituirá o componente mecânico.

Figura 18 – Tela de caracterização linear do material do *Multiscale Designer*, da Altair. Em (A) são escolhidos os materiais da matriz e da fibra (*Phase 1* e *Phase 2*), a partir da base de dados de materiais acessada em (B). Em (C) as várias camadas de fibras são estruturadas a partir de seus ângulos, resultando em 4 compostos diferentes. Finalmente em (D) temos os resultados da simulação linear do material.



Fonte: Criada pelo autor.

Figura 19 – Representação ilustrativa da configuração de fibras na estrutura laminada com fibras. Da esquerda para a direita tem-se o RVE, o material em macro-escala, o material laminado e finalmente o componente mecânico.



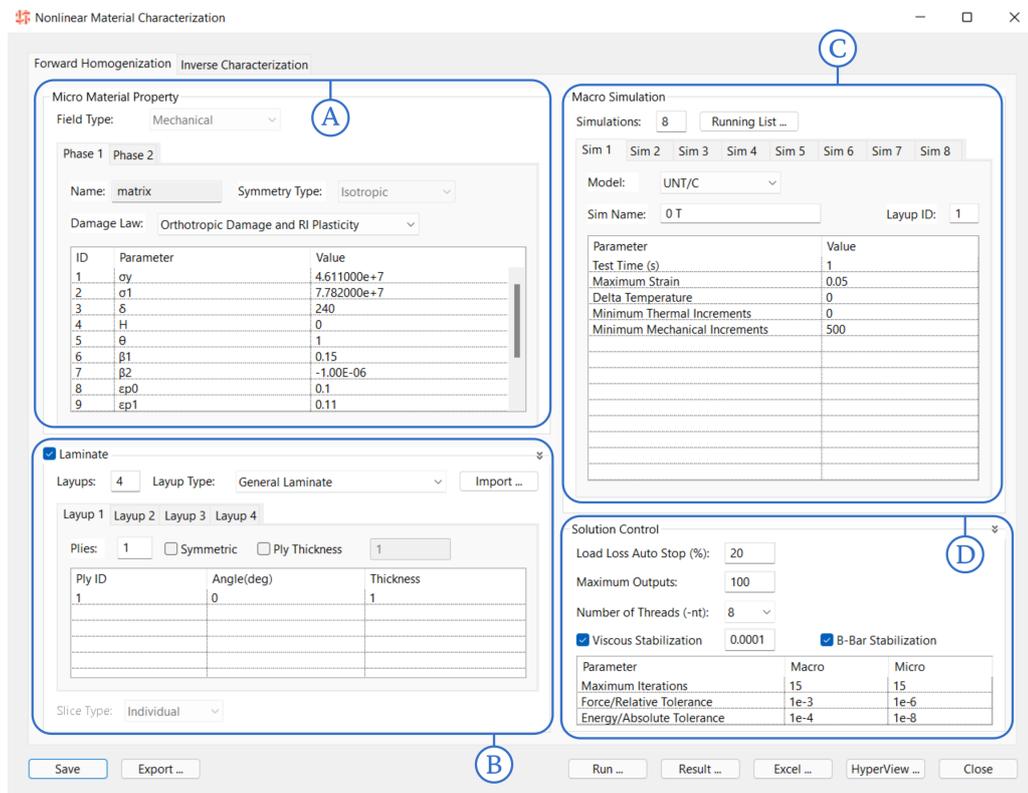
Fonte: Adaptado de Omairey, Dunning e Sriramula (2019)

4.4 Caracterização não linear do material

Após o término bem sucedido da simulação linear, e respectivo sinal verde  na tela da Figura 15, a opção de caracterização não linear do material torna-se disponível. Na Figura 20

tem-se as propriedades dos materiais já definidas em (A), assim como as estruturas laminadas já propostas em (B), de forma similar à da Figura 18. Em (C) tem-se a área de macrosimulação (*Macro Simulation*), onde podem ser criadas simulações para os materiais já gerados.

Figura 20 – Tela de caracterização não linear do material do *Multiscale Designer*, da Altair.



Fonte: Criada pelo autor.

Foram criadas 8 simulações, duas para cada tipo de material, sendo uma um ensaio de compressão, e a outra um ensaio de tração. Nessas simulações foram definidos o tempo de teste de 1 s (*Test time (s)*), padrão para todas, a deformação máxima de 0,05 para casos de tração, e -0,05 para casos de compressão (*Maximum Strain*), variação de temperatura (*Delta Temperature*) e o incremento termal mínimo (*Minimum Thermal Increments*) ambos em 0, e o incremento mecânico mínimo de 500 (*Minimum Mechanical Increments*). O controle da solução em (D) foi mantido em seus valores padrões do software para a máquina, exatamente como mostrado na Figura 20, para todos os materiais e simulações.

Esse procedimento foi repetido para 19 configurações diferentes quantidade de fibra, variando de 5 % a 50 %. Os resultados das simulações não lineares da tensão e deformação foram utilizados para o cálculo do módulo de Young do material na direção x em sua fase de deformação elástica, em Pa. A partir desse resultado, tem-se 152 módulos de Young diferentes.

4.5 Modelo de redes neurais

Esses dados foram organizados de acordo com sua configuração, tanto em porcentagem de fibras, como em estrutura laminada, sendo esse último um dado categórico, tratado com uma técnica denominada *one-hot-encoding* (OKADA; OHZEKI; TAGUCHI, 2019). Toda a manipulação dos dados foi feita com o uso da biblioteca Pandas (MCKINNEY, 2010).

Todos esses dados foram então randomizados e normalizados, utilizando-se do escalador de máximo absoluto no intervalo $[0, 1]$, `preprocessing.MaxAbsScaler`¹ do Scikit-Learn (PEDREGOSA *et al.*, 2011), em cada característica separadamente, e então divididos em três diferentes conjuntos: dados de treino, dados de validação, e dados de teste. Inicialmente, todos os dados foram separados aleatoriamente em dados de treino e de teste, na proporção de 90 % para treino, e 10 % para teste. Os dados de validação correspondem a 25 % dos dados de treino, também escolhidos aleatoriamente. A separação aleatória dos dados foi obtida com o método `DataFrame.sample`² do Pandas.

Cada conjunto de dados compreendia 6 características: 5 de entrada — uma para cada estrutura laminada e uma para a porcentagem de fibra — e uma de saída, alvo da rede neural, o módulo de Young.

Várias redes neurais densas com topologias diferentes foram treinadas para encontrar qual seria a que melhor se adequaria aos dados. Todas possuíam camada de entrada com 5 neurônios e camada de saída de 1 neurônio. Inicialmente foram testadas redes com as mesmas quantidades de camadas e de neurônios, alterando a função de ativação. Definida a melhor função de ativação, testou-se diversas combinações de hiperparâmetros (nesse caso, número de camadas ocultas e de neurônios por camada oculta), para encontrar a melhor rede possível. Essas redes podem ser encontradas na Tabela 2, com seus otimizadores e métricas na Tabela 3.

Tabela 2 – Topologia das redes neurais testadas para previsão do módulo de Young. As camadas de entrada e de saída possuem 5 e 1 neurônios, respectivamente, para todos os modelos.

Modelo	Camadas ocultas	Neurônios	Função de ativação
1	28	40	reLU
2	28	40	sigmoid
3	28	40	tanh
4	68	200	reLU
5	78	300	reLU
6	88	300	reLU
7	98	300	reLU

Fonte: Criada pelo autor.

¹ <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MaxAbsScaler.html>

² <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sample.html>

Tabela 3 – Otimizador e métricas utilizadas nas redes

Otimizador	Adam, taxa de aprendizado de 10^{-6} com <i>asmgrad</i> Reddi, Kale e Kumar (2019)	
Função de perda	Erro médio absoluto percentual	$MAPE = \frac{100\%}{n} \sum_i^n (y_i - \hat{y}_i)/y_i $
Métricas	Erro médio absoluto	$MAE = \frac{1}{n} \sum_i^n y_i - \hat{y}_i $
	Erro médio quadrático	$MSE = \frac{1}{n} \sum_i^n (y_i - \hat{y}_i)^2$

Fonte: Criada pelo autor.

As redes foram criadas com base nas bibliotecas Keras (CHOLLET *et al.*, 2015) e TensorFlow (MARTÍN ABADI *et al.*, 2015). A criação do Modelo 1 da Tabela 2 pode, a título de exemplo, ser implementada com essas bibliotecas na forma da Listagem 4.1. Com as redes treinadas, foram obtidos novos resultados de alta fidelidade com o uso do software *Multiscale Designer*, com porcentagens de fibra de 3 %, 22 %, 47 % e 55 %, utilizando do mesmo procedimento já citado, permitindo a comparação da previsão dos modelos bem sucedidos no treinamento com esses novos resultados de alta fidelidade, inéditos para as redes neurais. Com as redes treinadas e testadas com resultados de a alta fidelidade, o modelo com melhor resultado comparativo foi utilizado para prever resultados experimentais obtidos em Marlett (2011).

As configurações da máquina e as versões dos softwares utilizados no desenvolvimento desse trabalho podem ser encontrados na Tabela 4.

Tabela 4 – Configurações da máquina, e versões dos softwares utilizados.

Máquina	
Processador	Intel® Core™ i5-9300H CPU @ 2.40GHZ ×4
Memória	8 GB DDR4 2666 MHz
Placa de Vídeo	NVIDIA GeForce GTX 1050 3GB Max-Q
Sistema Operacional Microsoft Windows, versão 11	
Multiscale Designer	2021.1.0
Sistema Operacional GNU/Linux Mint, kernel 5.4.0-120-generic	
Python	3.8.10
Numpy	1.17.4
TensorFlow	2.7.0
Pandas	1.3.5
SciKit-Learn	1.0.1

Fonte: Criada pelo autor.

Listagem 4.1 – Exemplo de implementação em Python da construção de uma rede neural com 30 camadas. As camadas de entrada e saída são densas com 5 e 1 neurônio, respectivamente. As camadas ocultas intermediárias possuem 40 neurônios cada. Todas utilizam como função de ativação a relu. O modelo final é construído com uma função de perda MAPE, otimizador Adam e métricas de convergência MAE, MSE e MAPE.

```
1     # Carregando as bibliotecas TensorFlow e Keras
2     import tensorflow as tf
3     from tensorflow import keras
4
5     # Construindo a lista de camadas.
6     #
7     # A primeira é do tipo densa (Dense), com 5 neurônios e função de
8     # ativação relu
9     layers = [keras.layers.Dense(5, activation='relu', input_shape=len(data))]
10
11    # Loop para construção das camadas ocultas.
12    #
13    # São adicionadas à lista 28 camadas densas (Dense), com 40 neurônios cada,
14    # com função de ativação relu.
15    for c in range(28):
16        layers.extend(keras.layers.Dense(40, activation='relu'))
17
18    # Camada final, de saída.
19    #
20    # Construída de forma análoga às anteriores, com um único neurônio.
21    layers.extend(keras.layers.Dense(1, activation='relu'))
22
23    # Criando o modelo da rede com função de perda MAPE, Adam como otimizador
24    # e usando as métricas MAE, MSE e MAPE (cf. Tabela 3) para avaliação
25    # de convergência.
26    model = keras.Sequential(layers)
27    model.compile(loss='mape',
28                  optimizer=tf.keras.optimizers.Adam(learning_rate=1e-6, amsgrad=True),
29                  metrics=['mae', 'mse', 'mape'])
```

Fonte: Criada pelo autor.

5 Resultados e discussão

5.1 Resultados

Nessa seção serão apresentados os resultados comparativos de cada rede testada, bem como o desempenho de cada modelo comparativamente às simulações de alta fidelidade do *Multiscale Designer*. O melhor modelo, ou seja, aquele com as menores métricas de erro, será testado com resultados experimentais.

As várias topologias para a rede neural descritas no Capítulo 4, foram avaliadas de acordo com a função de perda e as métricas de erro da Tabela 3 no conjunto de dados de teste, além do coeficiente de determinação (CHICCO; WARRENS; JURMAN, 2021) dado por

$$R^2 = 1 - \frac{\sum \hat{y}_i - y_i}{\sum \bar{y}_i - y_i},$$

com \hat{y}_i , y_i e \bar{y}_i são, respectivamente, a i -ésima predição, o i -ésimo dado real e a média dos dados reais.

O resultado dessa avaliação pode ser apreciado na Tabela 5. Os gráficos com a relação das métricas em relação as épocas para todas as redes podem ser encontradas no Apêndice A

Tabela 5 – Erros médios absoluto, quadrático e absoluto percentual — respectivamente MAE, MSE e MAPE — e coeficiente de determinação R^2 das redes treinadas.

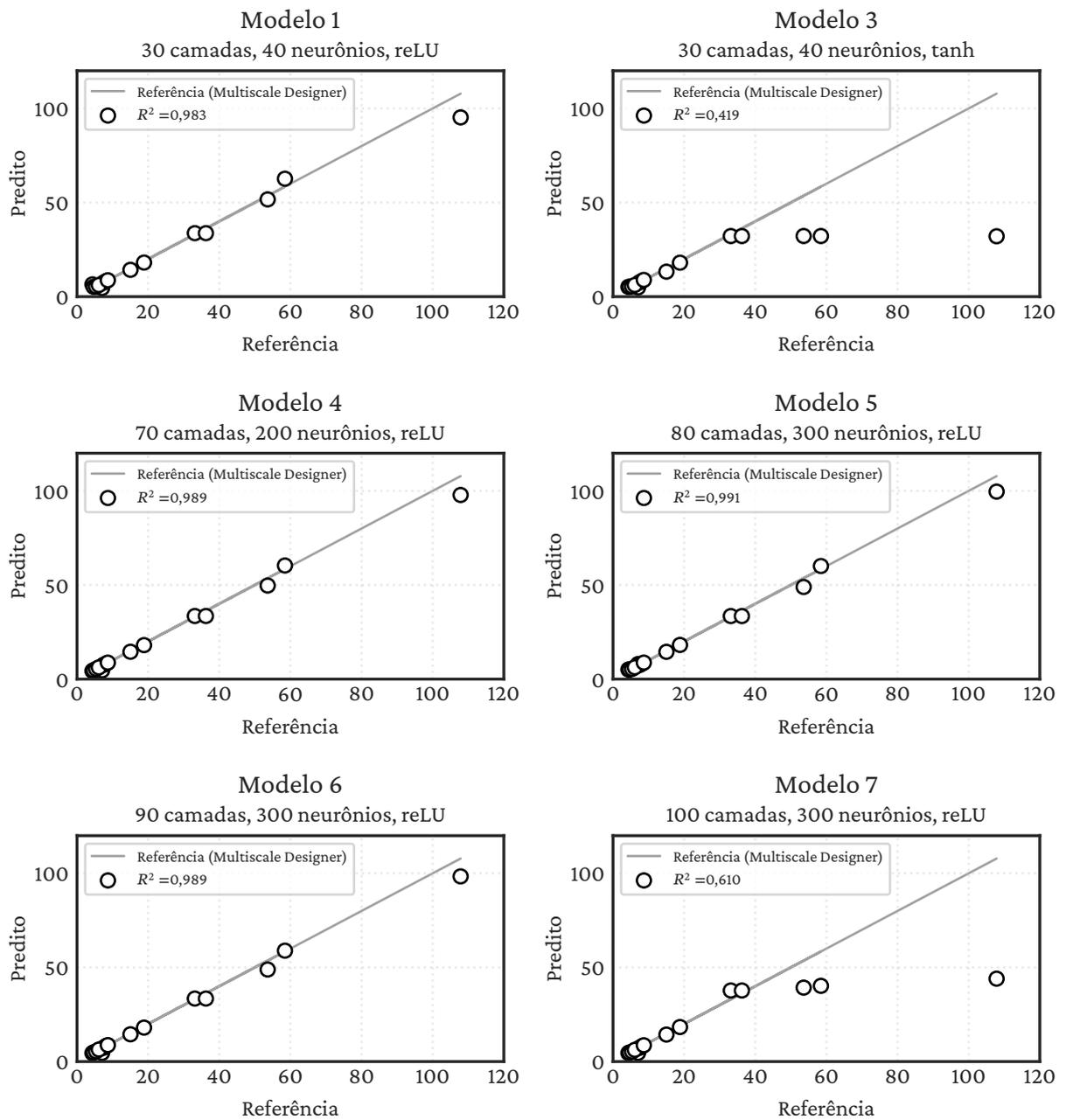
Modelo	Camadas	Neurônios	Ativação	MAE	MSE	MAPE (%)	R^2
1	30	40	reLU	$1,70 \times 10^{-2}$	$1,10 \times 10^{-3}$	9,27	0,983
2	30	40	sigmoid	$16,69 \times 10^{-2}$	$88,92 \times 10^{-3}$	49,30	---
3	30	40	tanh	$8,03 \times 10^{-2}$	$36,61 \times 10^{-3}$	17,05	0,419
4	70	200	reLU	$1,37 \times 10^{-2}$	$0,71 \times 10^{-3}$	5,15	0,989
5	80	300	reLU	$1,26 \times 10^{-2}$	$0,54 \times 10^{-3}$	5,21	0,991
6	90	300	reLU	$1,34 \times 10^{-2}$	$0,67 \times 10^{-3}$	5,76	0,989
7	100	300	reLU	$6,35 \times 10^{-2}$	$24,59 \times 10^{-3}$	12,71	0,610

Dos resultados dos modelos 1, 2 e 3, nota-se que entre as redes de mesma quantidade de neurônios e de camadas, a que possui a função de ativação reLU é a que se destaca com o menor valor em todas as métricas, ou seja, a que permite que a rede melhor se adapte ao problema. Assim, a função de ativação escolhida para novos testes será a reLU.

Aumentando tanto o número de camadas quanto de neurônios da rede, já com a função de ativação reLU fixada, nota-se que os modelos 4, 5 e 6 possuem resultados similares, enquanto o modelo 7 já não se adéqua tão bem aos dados de treinamento. Isso mostra que existe um limite de número de camadas e neurônios, que se ultrapassado pode resultar em aumento das métricas de erro, o que pode caracterizar o *overfitting* (GOODFELLOW; BENGIO; COURVILLE, 2016; CHOLLET, 2017).

Os gráficos apresentados na Figura 21 mostram a correlação do resultado de cada modelo com o resultado esperado pela simulação de alta fidelidade nos dados de teste. No eixo das abscissas tem-se os resultados de referência de alta fidelidade gerados pelo *Multiscale Designer*. A reta representa a correlação perfeita ($R^2 = 1$), enquanto no eixo das ordenadas tem-se os resultados preditos pelo modelo de rede neural. Nesse caso, é impossível distinguir o que é compressão ou tração, devido à aleatoriedade na geração do conjunto de testes.

Figura 21 – Gráfico de correlação nos dados de teste. O eixo das abscissas traz os resultados de alta fidelidade do *Multiscale Designer*, o das ordenadas são os valores preditos pela rede neural e a reta representa a correlação perfeita ($R^2 = 1$).



Fonte: Criada pelo autor.

O modelo 2, que utiliza a função de ativação sigmoide não aparece na Figura 21 pois foi

incapaz de construir a correlação entre a previsão e os dados originais. Assim como mostrado na Tabela 5 as redes 1, 4, 5 e 6 são as que trazem os melhores resultados.

Todos os modelos foram treinados com dados de porcentagem de fibra entre 5 % e 50 %; então, para testar suas capacidades preditivas e extrapolativas, gerou-se um novo conjunto de dados no *Multiscale Designer*. Os valores escolhidos para as novas configurações foram de: 3 %, 22 %, 47 % e 55 %, de porcentagem de fibra, nas quatro diferentes estruturas de camadas. Dessa forma produziu-se dois valores fora do intervalo de porcentagem de fibra apresentado às redes (um menor e um maior), e dois dentro desse intervalo.

Os gráficos apresentados nas Figuras 22 e 23 mostram a correlação do resultado de cada modelo com o resultado esperado pela simulação de alta fidelidade nesses novos dados. Nesse caso é possível separar entre ensaios de compressão e tração, pois se conhece cada dado separadamente, mesmo que inéditos para rede. No eixo das abscissas tem-se os resultados de referência de alta fidelidade gerados pelo *Multiscale Designer*. A reta representa a correlação perfeita ($R^2 = 1$), enquanto no eixo das ordenadas tem-se os resultados preditos pelo modelo de rede neural.

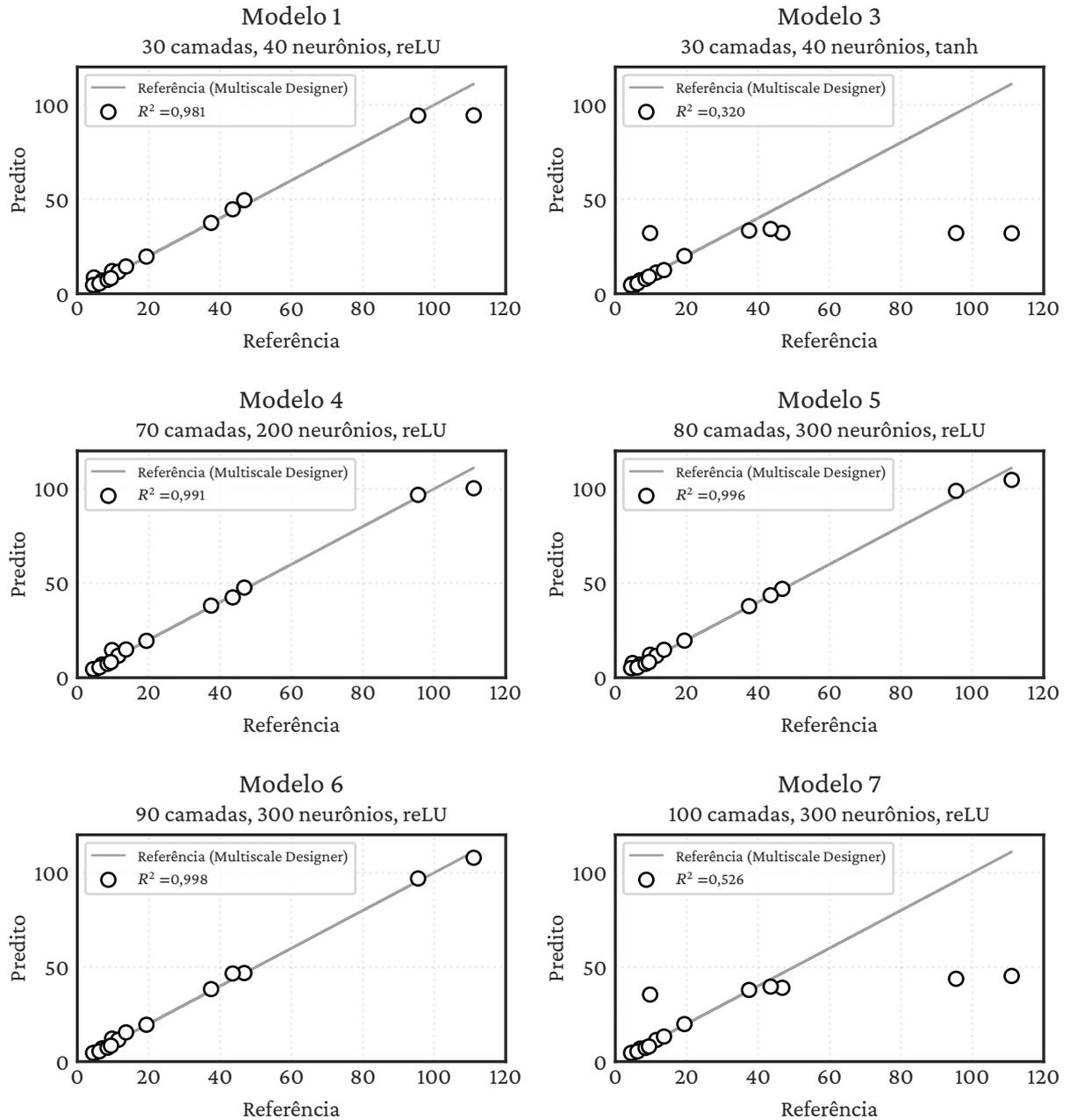
Pelos resultados de correlação R^2 nota-se que as redes com função de ativação reLU são as melhores para esse tipo de problema, sendo que os modelos os modelos 1, 4, 5 e 6 são aqueles que se destacam. Utilizando a função de ativação sigmoide, o modelo 2 não aparece nas Figuras 22 e 23 pois produziu apenas resultados constantes, sendo incapaz de generalizar os dados de treinamento. No entanto o modelo 7, mesmo utilizando da mesma função de ativação apresenta uma ótima correlação para alguns pontos do domínio, principalmente os de menor valor e erra em valores maiores. Uma das explicações para esse comportamento é o sobreajuste (*overfitting*), ou seja, por ter parâmetros demais a rede se ajustou mais do que o necessário nos dados de treino, a ponto de não se adaptar a novos dados, não sendo consistente para resolver o problema de forma ampla. Outra possível explicação é uma incapacidade de ajuste da rede, pois pelos gráficos do processo de treinamento mostrados no Apêndice A, o erro de validação torna-se menor que o de treinamento, o que só faria sentido se os dados de treinamento fossem muito complexos, ou se os de validação muito simples; por ser a única rede a apresentar esse comportamento, esse fato não é provável.

A comparação dos resultados do módulo de Young dos modelos bem sucedidos (indicados por E_p) com a simulação de alta fidelidade usando o *Multiscale Designer* (E_r) pode ser encontrada nas tabelas a seguir, junto do erro percentual $100\%(E_r - E_p)/E_r$. Nas Tabelas 6 e 7 são apresentados os resultados para o modelo 4, do modelo 5 nas Tabelas 8 e 9 e do modelo 6 nas Tabelas 10 e 11.

Os erros relativos médios dos dados de previsão em relação ao original podem ser encontrados por modelo e por ensaio na Tabela 12.

Novamente, nota-se que os três modelos possuem resultados muito próximos que, em média, são precisos. No entanto, existem erros relativos altos em algumas das redes, como no modelo 5, onde para o caso de 3 % de fibra e na configuração quase-isotrópica no ensaio de compressão o erro chega a quase 60 %; no entanto, esse erro acaba sendo menos relevante de

Figura 22 – Gráfico de correlação para um ensaio de compressão. O eixo das abscissas traz os resultados de alta fidelidade do *Multiscale Designer*, o das ordenadas são os valores preditos pela rede neural e a reta representa a correlação perfeita ($R^2 = 1$).

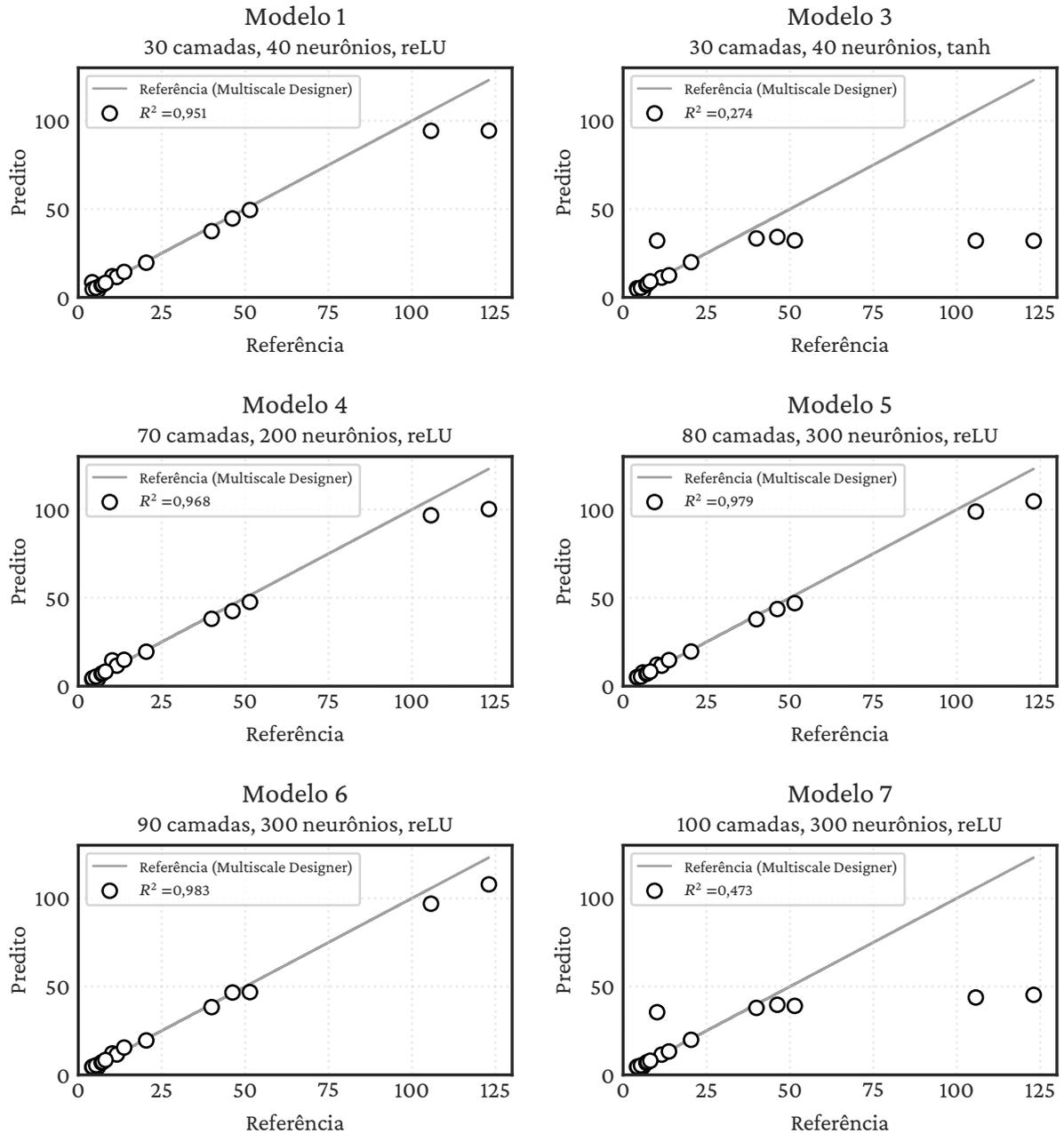


Fonte: Criada pelo autor.

maneira absoluta pois, como já esperado, os casos de 3% de fibra são os com menores módulos de Young.

É importante observar que os maiores erros das redes encontram-se nos dados em que é necessária uma extrapolação, ou seja, em 3% e 55% de fibra. Logo, o aprendizado de máquina funciona dentro do esperado, com bons resultados no alcance dos dados de treino, e resultados piores fora desses limites. Mesmo assim bons resultados foram alcançados nos casos de extrapo-

Figura 23 – Gráfico de correlação para um ensaio de tração. O eixo das abscissas traz os resultados de alta fidelidade do *Multiscale Designer*, o das ordenadas são os valores preditos pela rede neural e a reta representa a correlação perfeita ($R^2 = 1$).



Fonte: Criada pelo autor.

lações, como erros de 2,3 % no modelo 4, para o caso de 3 % de fibra e na configuração $45^\circ/-45^\circ$. De fato, a configuração de $45^\circ/-45^\circ$ é a que possui os menores erros percentuais, obtendo bons resultados até mesmo nas extrapolações em questão de porcentagem de fibra.

Entretanto, um fato a se atentar em todos os modelos e ensaios é que a rede prevê exatamente o mesmo valor para compressão e para tração, pois esse não é um dado de entrada, logo todos os modelos aprenderam a calcular uma média, que se adequa aos valores, independente-

Tabela 6 – Erros percentuais para o ensaio de compressão para cada uma das porcentagens de fibra e configuração de camadas, para o modelo 4 (70 camadas, 200 neurônios, reLU). O módulo de Young está em GPa.

Camadas	0°			45°/-45°			90°			Quase-isotrópica		
	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$
3	9,75	14,61	49,9	4,42	4,53	2,3	4,65	3,98	-14,4	4,86	4,22	-13,3
22	46,77	47,65	1,9	6,97	7,00	0,4	6,21	5,46	-12,1	19,37	19,49	0,6
47	95,45	96,73	1,3	11,49	11,60	0,9	8,48	7,35	-13,4	37,48	38,11	1,7
55	111,03	100,23	-9,7	13,65	14,92	9,3	9,43	8,25	-12,5	43,52	42,47	-2,4

Fonte: Criada pelo autor.

Tabela 7 – Erros percentuais para o ensaio de tração para cada uma das porcentagens de fibra e configuração de camadas, para o modelo 4 (70 camadas, 200 neurônios, reLU). O módulo de Young está em GPa.

Camadas	0°			45°/-45°			90°			Quase-isotrópica		
	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$
3	10,20	14,61	43,3	4,33	4,53	4,6	4,18	3,98	-4,6	5,92	4,22	-28,8
22	51,43	47,65	-7,4	6,99	7,00	0,1	5,38	5,46	1,4	20,37	19,49	-4,3
47	105,67	96,73	-8,5	11,56	11,60	0,3	7,31	7,35	0,5	39,96	38,11	-4,6
55	123,03	100,23	-18,5	13,74	14,92	8,6	8,14	8,25	1,3	46,21	42,47	-8,1

Fonte: Criada pelo autor.

Tabela 8 – Erros percentuais para o ensaio de compressão para cada uma das porcentagens de fibra e configuração de camadas, para o modelo 5 (80 camadas, 300 neurônios, reLU). O módulo de Young está em GPa.

Camadas	0°			45°/-45°			90°			Quase-isotrópica		
	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$
3	9,75	12,13	24,4	4,42	5,08	14,8	4,65	5,08	9,1	4,86	7,77	59,7
22	46,77	46,98	0,5	6,97	6,99	0,4	6,21	5,42	-12,7	19,37	19,60	1,2
47	95,45	98,84	3,5	11,49	11,57	0,7	8,48	7,34	-13,5	37,48	37,84	1,0
55	111,03	104,65	-5,7	13,65	14,75	8,1	9,43	8,26	-12,4	43,52	43,62	0,2

Fonte: Criada pelo autor.

Tabela 9 – Erros percentuais para o ensaio de tração para cada uma das porcentagens de fibra e configuração de camadas, para o modelo 5 (80 camadas, 300 neurônios, reLU). O módulo de Young está em GPa.

Camadas	0°			45°/-45°			90°			Quase-isotrópica		
	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$
3	10,20	12,13	18,9	4,33	5,08	17,4	4,18	5,08	21,6	5,92	7,77	31,1
22	51,43	46,98	-8,7	6,99	6,99	0,0	5,38	5,42	0,7	20,37	19,60	-3,8
47	105,67	98,84	-6,5	11,56	11,57	0,1	7,31	7,34	0,4	39,96	37,84	-5,3
55	123,03	104,65	-14,9	13,74	14,75	7,4	8,14	8,26	1,4	46,21	43,62	-5,6

Fonte: Criada pelo autor.

mente do tipo de deformação, o que poderia justificar a facilidade de aprender a configuração 45°/-45°, pois essa possui módulos de Young muito similares em ambos os casos.

Para corrigir esse viés, foram treinados novos modelos com variáveis categóricas extras

Tabela 10 – Erros percentuais para o ensaio de compressão para cada uma das porcentagens de fibra e configuração de camadas, para o modelo 6 (90 camadas, 300 neurônios, reLU). O módulo de Young está em GPa.

Camadas	0°			45°/-45°			90°			Quase-isotrópica		
Porcentagem de fibra	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$
3	9,75	12,17	24,9	4,42	4,61	4,1	4,65	4,47	-3,9	4,86	4,56	-6,3
22	46,77	46,87	0,2	6,97	7,04	1,0	6,21	5,47	-11,9	19,37	19,50	0,7
47	95,45	96,89	1,5	11,49	11,56	0,6	8,48	7,35	-13,4	37,48	38,37	2,4
55	111,03	107,86	-2,8	13,65	15,48	13,4	9,43	8,46	-10,3	43,52	46,64	7,2

Fonte: Criada pelo autor.

Tabela 11 – Erros percentuais para o ensaio de tração para cada uma das porcentagens de fibra e configuração de camadas, para o modelo 6 (90 camadas, 300 neurônios, reLU). O módulo de Young está em GPa.

Camadas	0°			45°/-45°			90°			Quase-isotrópica		
Porcentagem de fibra	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$
3	10,20	12,17	19,4	4,33	4,61	6,5	4,18	4,47	7,2	5,92	4,56	-23,1
22	51,43	46,87	-8,9	6,99	7,04	0,7	5,38	5,47	1,7	20,37	19,50	-4,3
47	105,67	96,89	-8,3	11,56	11,56	0,0	7,31	7,35	0,5	39,96	38,37	-4,0
55	123,03	107,86	-12,3	13,74	15,48	12,6	8,14	8,46	3,9	46,21	46,64	0,9

Fonte: Criada pelo autor.

Tabela 12 – Erros relativos médios dos modelos nos dados inéditos em %.

Modelo	Ensaio de tração	Ensaio de compressão
4	9,06	9,14
5	8,99	10,49
6	7,13	6,53

que discriminavam para a rede se era o caso de treino era de tração ou de compressão. Os modelos treinados se assemelham aos modelos 4, 5 e 6, com a exceção da camada de entrada, agora com 7 neurônios para acomodar os novos dados. A avaliação dessas redes teve como resultado métricas e função de perda segundo a Tabela 13.

Tabela 13 – Erros médios absoluto, quadrático e absoluto percentual — respectivamente MAE, MSE e MAPE — das redes treinadas, agora incluindo um dado categórico para discriminação dos ensaios de compressão e tensão.

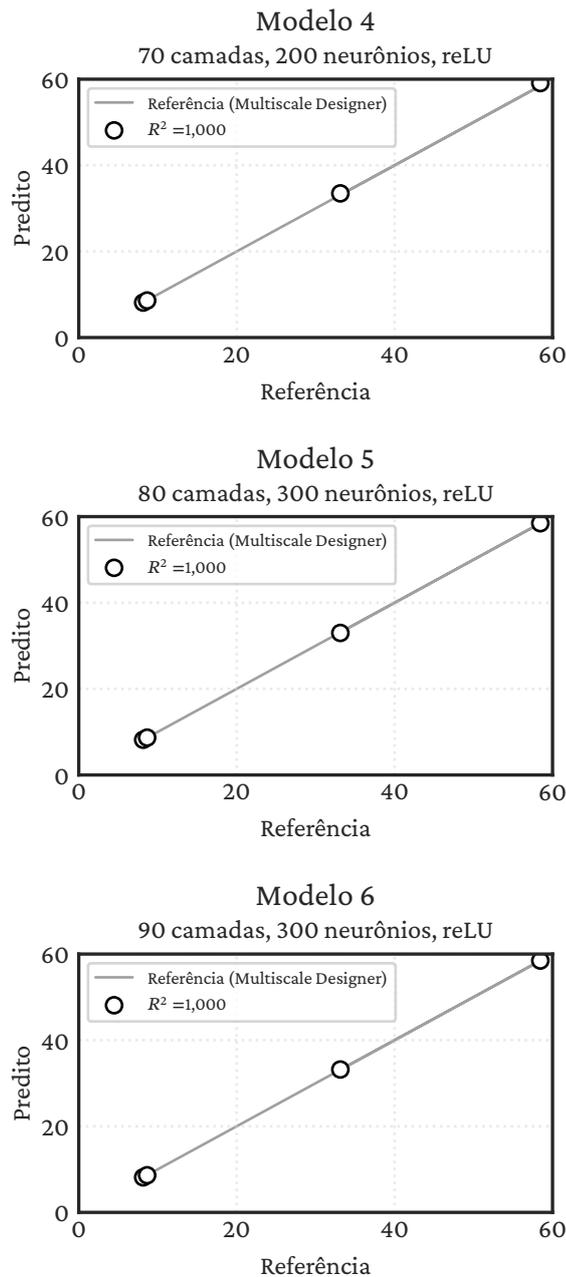
Modelo	Camadas	Neurônios	Ativação	MAE	MSE	MAPE (%)
4	70	200	reLU	$6,6 \times 10^{-3}$	$2,24 \times 10^{-4}$	2,54
5	80	300	reLU	$3,7 \times 10^{-3}$	$5,95 \times 10^{-5}$	1,81
6	90	300	reLU	$1,5 \times 10^{-3}$	$1,51 \times 10^{-5}$	2,07

Novamente é possível notar uma similaridade muito grande entre os modelos, possuindo funções de perda (MAPE) muito similares.

É possível calcular o R^2 nos dados de teste, separadamente para tensão e compressão,

nesse caso, devido a variável categórica, e tais resultados podem ser avaliados nos gráficos de correlação das Figuras 24 e 25, e por isso esses não se encontram na tabela 13.

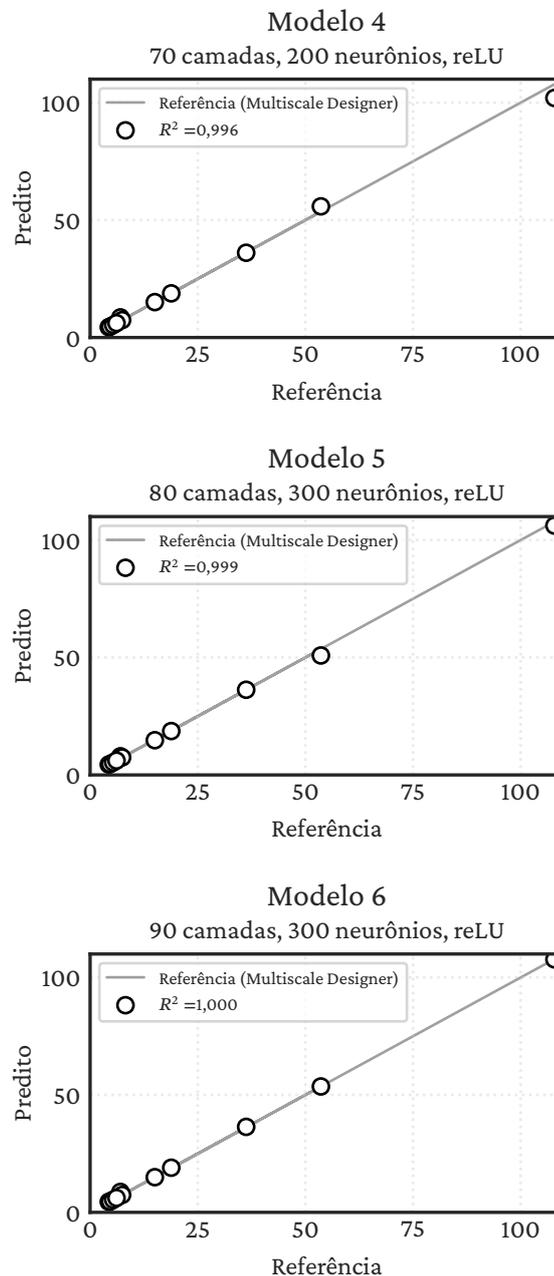
Figura 24 – Gráfico de correlação para um ensaio de compressão após a introdução de um dado categórico que discrimina entre problemas de compressão e de tração nos dados de teste. O eixo das abscissas traz os resultados de alta fidelidade do *Multiscale Designer*, o das ordenadas são os valores preditos pela rede neural e a reta representa a correlação perfeita ($R^2 = 1$).



Fonte: Criada pelo autor.

Nota-se que para os dados de teste a correlação é quase perfeita, tendo $R^2 = 1$ em todos os casos de compressão, e apenas um resultado abaixo de 0.999 no ensaio de tração. Mostrando que a rede com esse dado categórico já apresenta melhores resultados.

Figura 25 – Gráfico de correlação para um ensaio de tração após a introdução de um dado categórico que discrimina entre problemas de compressão e de tração nos dados de teste. O eixo das abscissas traz os resultados de alta fidelidade do *Multiscale Designer*, o das ordenadas são os valores preditos pela rede neural e a reta representa a correlação perfeita ($R^2 = 1$).

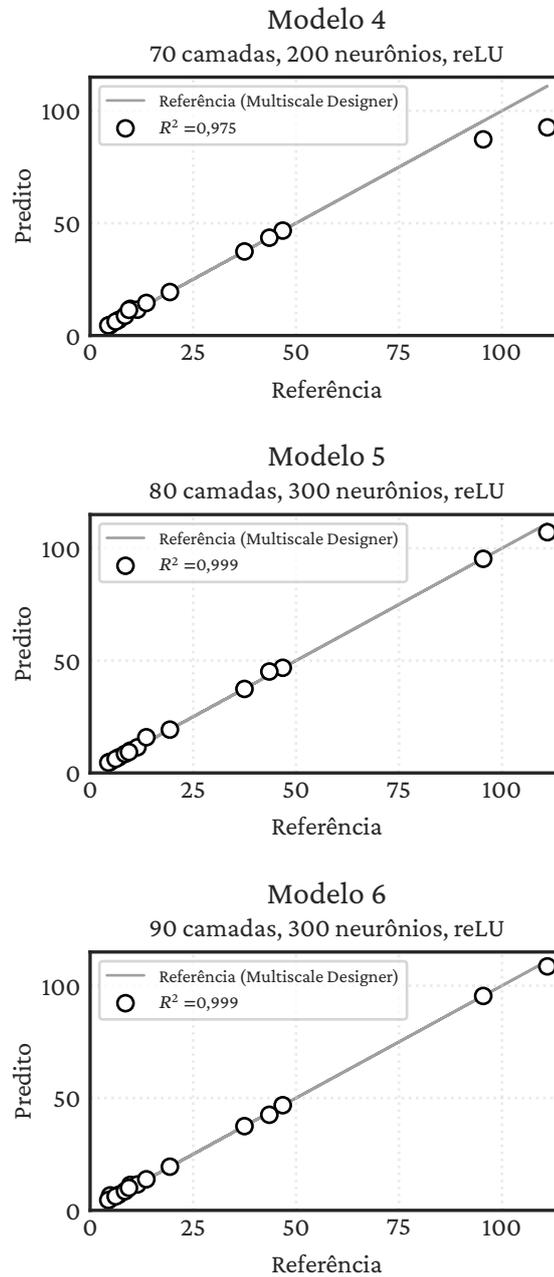


Fonte: Criada pelo autor.

Todos os novos modelos foram testados com os dados inéditos anteriormente descritos, para testar sua capacidade de predição e extrapolação em novos dados, resultando nos gráficos de correlação das Figuras 26 e 27.

A partir dos gráficos conclui-se que as três redes treinadas têm novamente bons resultados, todas obtendo $R^2 \geq 0,99$. Gráficamente pode-se perceber que o modelo 6 é o que melhor se adequa

Figura 26 – Gráfico de correlação para um ensaio de compressão após a introdução de um dado categórico que discrimina entre problemas de compressão e de tração. O eixo das abscissas traz os resultados de alta fidelidade do *Multiscale Designer*, o das ordenadas são os valores preditos pela rede neural e a reta representa a correlação perfeita ($R^2 = 1$).

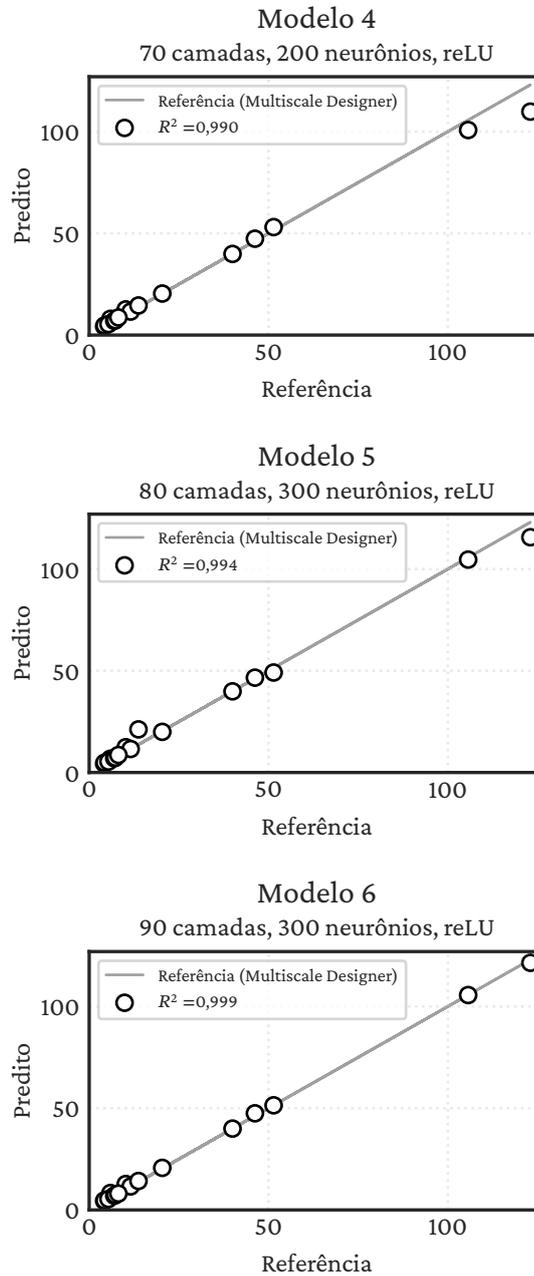


Fonte: Criada pelo autor.

aos dados de maneira geral, fato confirmado pelo $R^2 = 0,999$ em ambos os ensaios, tração e compressão.

A comparação dos resultados do módulo de Young dos modelos 4, 5 e 6 (indicados por E_p) com a simulação de alta fidelidade usando o *Multiscale Designer* (E_r) pode ser encontrada nas tabelas a seguir, junto do erro percentual $100\%(E_r - E_p)/E_r$. Nas Tabelas 14 e 15 são apresentados

Figura 27 – Gráfico de correlação para um ensaio de tração após a introdução de um dado categórico que discrimina entre problemas de compressão e de tração. O eixo das abscissas traz os resultados de alta fidelidade do *Multiscale Designer*, o das ordenadas são os valores preditos pela rede neural e a reta representa a correlação perfeita ($R^2 = 1$).



Fonte: Criada pelo autor.

os resultados para o modelo 4, do modelo 5 nas Tabelas 16 e 17 e do modelo 6 nas Tabelas 18 e 19.

Mais uma vez os maiores erros se encontram nas frações de fibra de 3% e de 55% em todos os modelos, o que é esperado, devido a ser uma extrapolação.

Os erros relativos médios dos dados de previsão em relação ao original podem ser encontrados por modelo e por ensaio na tabela 20.

Tabela 14 – Erros percentuais para o ensaio de compressão para cada uma das porcentagens de fibra e configuração de camadas, para o modelo 4 (70 camadas, 200 neurônios, reLU). O módulo de Young está em GPa.

Camadas	0°			45°/-45°			90°			Quase-isotrópica		
Porcentagem de fibra	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$
3	9,75	11,89	22,0	4,42	4,49	1,4	4,65	4,58	-1,5	4,86	4,63	-4,9
22	46,77	46,70	-0,1	6,97	6,97	-0,0	6,21	6,21	-0,0	19,37	19,35	-0,1
47	95,45	87,30	-8,5	11,49	11,49	-0,0	8,48	8,78	3,5	37,48	37,41	-0,2
55	111,03	92,65	-16,6	13,65	14,49	6,2	9,43	11,41	21,0	43,52	43,54	0,0

Tabela 15 – Erros percentuais para o ensaio de tração para cada uma das porcentagens de fibra e configuração de camadas, para o modelo 4 (70 camadas, 200 neurônios, reLU). O módulo de Young está em GPa.

Camadas	0°			45°/-45°			90°			Quase-isotrópica		
Porcentagem de fibra	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$
3	10,20	12,62	23,7	4,33	4,73	9,3	4,18	4,29	2,6	5,92	7,98	34,7
22	51,43	53,07	3,2	6,99	6,99	-0,1	5,38	5,38	-0,1	20,37	20,41	0,2
47	105,67	100,79	-4,6	11,56	11,55	-0,1	7,31	7,30	-0,1	39,96	39,88	-0,2
55	123,03	109,84	-10,7	13,74	14,55	5,9	8,14	8,67	6,5	46,21	47,37	2,5

Tabela 16 – Erros percentuais para o ensaio de compressão para cada uma das porcentagens de fibra e configuração de camadas, para o modelo 5 (80 camadas, 300 neurônios, reLU). O módulo de Young está em GPa.

Camadas	0°			45°/-45°			90°			Quase-isotrópica		
Porcentagem de fibra	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$
3	9,75	9,92	1,8	4,42	4,58	3,6	4,65	4,87	4,6	4,86	4,89	0,5
22	46,77	46,80	0,1	6,97	6,96	-0,1	6,21	6,21	0,0	19,37	19,28	-0,5
47	95,45	95,30	-0,2	11,49	11,42	-0,6	8,48	8,48	-0,0	37,48	37,39	-0,2
55	111,03	107,16	-3,5	13,65	15,88	16,3	9,43	9,29	-1,4	43,52	45,14	3,7

Tabela 17 – Erros percentuais para o ensaio de tração para cada uma das porcentagens de fibra e configuração de camadas, para o modelo 5 (80 camadas, 300 neurônios, reLU). O módulo de Young está em GPa.

Camadas	0°			45°/-45°			90°			Quase-isotrópica		
Porcentagem de fibra	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$
3	10,20	12,43	21,9	4,33	4,86	12,2	4,18	4,42	5,8	5,92	6,77	14,4
22	51,43	49,14	-4,5	6,99	7,00	0,1	5,38	5,36	-0,3	20,37	19,98	-1,9
47	105,67	104,65	-1,0	11,56	11,55	-0,1	7,31	7,31	-0,1	39,96	39,89	-0,2
55	123,03	115,64	-6,0	13,74	21,20	54,3	8,14	8,61	5,7	46,21	46,61	0,9

Tabela 18 – Erros percentuais para o ensaio de compressão para cada uma das porcentagens de fibra e configuração de camadas, para o modelo 6 (90 camadas, 300 neurônios, reLU). O módulo de Young está em GPa.

Camadas	0°			45°/-45°			90°			Quase-isotrópica		
Porcentagem de fibra	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$
3	9,75	11,34	16,3	4,42	4,59	3,8	4,65	4,79	2,9	4,86	6,66	36,9
22	46,77	46,80	0,1	6,97	6,96	-0,1	6,21	6,20	-0,1	19,37	19,43	0,3
47	95,45	95,44	-0,0	11,49	11,52	0,3	8,48	8,53	0,5	37,48	37,48	0,0
55	111,03	108,59	-2,2	13,65	13,83	1,3	9,43	10,00	6,1	43,52	42,54	-2,3

Tabela 19 – Erros percentuais para o ensaio de tração para cada uma das porcentagens de fibra e configuração de camadas, para o modelo 6 (90 camadas, 300 neurônios, reLU). O módulo de Young está em GPa.

Camadas	0°			45°/-45°			90°			Quase-isotrópica		
Porcentagem de fibra	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$	E_r	E_p	$\Delta\%$
3	10,20	12,73	24,8	4,33	4,68	8,1	4,18	4,48	7,4	5,92	8,30	40,1
22	51,43	51,43	0,0	6,99	7,00	0,1	5,38	5,38	-0,0	20,37	20,68	1,5
47	105,67	105,62	-0,1	11,56	11,56	0,0	7,31	7,32	0,1	39,96	39,98	0,0
55	123,03	121,47	-1,3	13,74	14,22	3,5	8,14	8,03	-1,4	46,21	47,49	2,8

Tabela 20 – Erros relativos médios dos modelos nos dados inéditos em %.

Modelo	Ensaio de tração	Ensaio de compressão
4	6,53	5,37
5	8,08	2,32
6	5,69	4,57

O destaque claro é para o modelo 5 em casos de compressão, pois apresenta um erro médio baixíssimo, com apenas um valor na ordem das dezenas, no caso de 55 % de fibra, o que é esperado devido ao fato de ser um dado fora dos limites de treino.

Já quando se analisa o modelo 6, nota-se que seus maiores erros estão nos 3 % de fibra, onde os valores do módulo de Young são menores, logo, mesmo com um erro relativo percentual médio mais alto, devido a valores da ordem de 40%, o valor absoluto continua próximo. Também é possível observar que nos dados que estão dentro do intervalo de treino o maior erro dessa rede é de 1,5 % mostrando uma grande facilidade em prever esse tipo de resultado.

Assim, confirma-se que o modelo 6 foi o que melhor se adequou aos dados de maneira geral, tanto para previsão dentro dos limites treinados, quanto para extrapolações, de maneira consistente em ambos ensaios. No entanto, isso não altera o fato que os modelos 4 e 5 ainda assim obtêm bons resultados, principalmente dentro do intervalo de treino, com destaque para o modelo 5 no ensaio de compressão.

Para testar a capacidade preditiva do modelo 6, seus resultados foram comparados com dados obtidos experimentalmente por Marlett (2011). Os resultados dessa comparação encontram-se nas Tabelas 21 e 22.

Tabela 21 – Ensaio de tração para comparação do módulo de Young (em GPa) entre o valor previsto pelo modelo 6 e o resultado experimental de Marlett (2011).

Configuração de Camadas	0°	90°	Quase-isotrópica
Porcentagem de fibra	59,56	58,72	57,13
Young Experimental	127,41	10,34	49,02
Young Previsto	129,80	8,39	49,96
Erro percentual $\Delta\%$	1,87	-18,84	1,31

Os dados do módulo de Young para a configuração de 45° não fazem parte da pesquisa

Tabela 22 – Ensaio de compressão para comparação do módulo de Young (em GPa) entre o valor previsto pelo modelo 6 e o resultado experimental de Marlett (2011).

Configuração de Camadas	0°	90°	Quase-isotrópica
Porcentagem de Fibra	61,76	61,48	58,42
Young Experimental	107,35	10,75	45,91
Young Previsto	114,78	11,73	44,03
Erro percentual $\Delta\%$	6,92	9,08	-4,09

de Marlett (2011). Entretanto, para as outras três configurações, têm-se resultados satisfatórios, com erro médio de 7,34 % no ensaio de tração e 6,97 % no ensaio de compressão. A rede continua com uma média de erro similar ao obtido em relação ao *Multiscale Designer*, aumentando cerca de 2 % em cada caso. Um erro baixo como esse indica uma ótima aproximação, mesmo com os dados utilizados para a porcentagem de fibra fora do alcance do treino da rede, ou seja, superiores a 50 %. Isso mostra não só uma adequação a uma simulação padrão ouro, como a do *Multiscale Designer*, como também a resultados experimentais.

Comparando os resultados da rede neural com o *Multiscale Designer* no quesito de tempo computacional, ambos executados na máquina descrita na Tabela 4, a rede neural traz a quantidade de respostas pedidas de maneira praticamente instantânea, demorando menos de um segundo para responder os resultados do módulo de Young para diversas configurações simultaneamente. Já o *Multiscale Designer*, mesmo que traga resultados mais completos, demora cerca de 20 minutos para gerar todos os dados de uma única configuração. No caso da pesquisa experimental os custos associados à realização dos testes em um material real são maiores do que apenas o tempo, que por si só já seria maior, devido a necessidade de se produzir em laboratório o material.

Neste trabalho as redes foram especificamente treinadas para aprender o módulo de Young em uma única dimensão, independente de essa ser longitudinal ou transversal. Para tanto o material foi considerado como elástico, durante a sua fase correspondente, com a deformação e estresse possuindo uma relação linear. A partir desses dados foi possível obter-se resultados muito satisfatórios comparados tanto aos de uma simulação de alta fidelidade, quanto a dados experimentais, em pouquíssimo tempo. No entanto, esse trabalho pode ser melhorado, a ponto de responder todas as outras propriedades do material, como por exemplo, o módulo de cisalhamento, o coeficiente de Poisson, o módulo de incompressibilidade e o módulo de rigidez.

6 Conclusão

Tendo em vista os aspectos observados até aqui, conclui-se que a utilização de redes neurais para a modelagem de propriedades elásticas de compósitos trata-se de um procedimento válido e promissor, permitindo obter resultados compatíveis com outros métodos, com um tempo muito menor.

O funcionamento de um modelo baseado em *Deep Learning* baseia-se em encontrar um melhor conjunto de pesos — a partir da escolha de uma métrica — que permitam à rede encontrar um mapa entre entradas e saídas. Isso fica claro no caso da rede aprender o módulo de Young sem uma distinção dos ensaios de tração e compressão, e em seguida com a informação categórica para ser aprendida. Ambos trazem resultados bons, mas sem oferecer todas as informações, a rede encontra uma resposta sub-ótima, buscando o melhor resultado possível baseada apenas naquilo que foi lhe apresentado, nesse caso uma média entre os valores do módulo de Young nos dois ensaios.

Como dito por Chollet (2017), a escolha da arquitetura certa de uma rede é mais próxima da arte que da Ciência. Logo, mesmo que existam motivos para imaginar que algum tipo específico de função de ativação possa ter bons resultados, ou uma noção da quantidade de parâmetros que a rede precisará, apenas com os testes computacionais é possível confirmar essas hipóteses. Entre as funções de ativações testadas, a que melhor se adequou ao caso da previsão do módulo de Young é a reLU, e os hiperparâmetros foram ajustados de maneira a se adequarem de maneira otimizada aos dados utilizados para treinamento, pois como mostrado, um grande número de camadas e neurônios pode resultar em uma rede com pior desempenho nos dados de teste e validação possivelmente devido ao *overfitting* nos dados de treino. Sendo assim, o modelo 6 foi a rede com os melhores resultados, construída com uma camada de entrada de 7 neurônios, 88 camadas ocultas com 300 neurônios, e uma camada de saída de 1 neurônio, com reLU como função de ativação.

Os modelos 4 e 5 também apresentaram bons resultados, podendo ser utilizados para o mesmo fim, caso seja necessário treinar uma rede com menos parâmetros devido à eficiência computacional.

De maneira geral nota-se que os modelos testados têm facilidade em encontrar os valores dentro do alcance dos dados de teste, e uma maior dificuldade em extrapolar esses limites. Esse fato era esperado, pois como explicado na Seção 2.3, um metamodelo baseado em aprendizado de máquina é construído em um conjunto finito e predeterminado de pontos, o que faz com que o ajuste funcione melhor dentro dos limites desses pontos do que na extrapolação. Mesmo assim, obtém-se bons resultados nesses casos, com a rede fornecendo resultados próximos tanto dos dados experimentais quanto dos obtidos por meio do *Multiscale Designer*.

Os resultados obtidos para o módulo de Young em ambos ensaios, previstos indepen-

dentemente de ser o módulo longitudinal ou transversal, são bons tanto quando comparados ao da simulação padrão ouro, quanto com os resultados experimentais, em diversas configurações laminadas, com a grande maioria dos erros relativos abaixo de 20 % para o modelo 6, e ainda executando de forma computacionalmente mais eficiente, mostrando que o aprendizado de máquina é uma opção viável para esse tipo de problema. Eliminando alguns dos limites mostrados na Seção 3.1 podem ser encontrados redes neurais que representem bem as propriedades elásticas, ao invés de explicitamente calcular modelos que podem ser limitados ou complexos.

Esse trabalho, mesmo que bem sucedido em sua proposta original ainda é limitado, estimando o módulo de Young em apenas uma direção durante a fase elástica de um compósito construído com apenas uma combinação de materiais. Portanto, ele pode ser aprimorado na direção da construção de modelos capazes de prever outras propriedades elásticas desse mesmo material, ou até mesmo de mais de um material, a partir da adição de mais variáveis de entrada e de saída para o treinamento.

Nesse caso, seria necessária uma nova avaliação de qual seria a melhor arquitetura possível para a rede, pois a aqui discutida traz bons resultados para essa função em específico, ou seja, deve-se avaliar quais funções de ativação, e quais combinações de hiperparâmetros se adequariam aos novos dados de entrada.

Referências

- ABOUDI, Jacob; ARNOLD, Steven M.; BEDNARCYK, Brett A. **Micromechanics of Composite Materials**. Edição: Jacob Aboudi, Steven M. Arnold e Brett A. Bednarcyk. Oxford: Butterworth-Heinemann, 2013. ISBN 978-0-12-397035-0. DOI: <https://doi.org/10.1016/B978-0-12-397035-0.00001-X>.
- ALTAIR ENGINEERING INC. **Altair® Multiscale Designer®**. 2021. Disponível em: <https://www.altair.com/multiscale-designer/>.
- ARIAN NIK, Mahdi *et al.* A comparative study of metamodeling methods for the design optimization of variable stiffness composites. **Composite Structures**, v. 107, p. 494–501, 2014. ISSN 0263-8223. DOI: <https://doi.org/10.1016/j.compstruct.2013.08.023>.
- BUNSELL, A.R.; JOANNÈS, S.; THIONNET, A. **Fundamentals of Fibre Reinforced Composite Materials**. CRC Press, 2021. (Series in Materials Science and Engineering). ISBN 9780429680946. Disponível em: <https://books.google.com.br/books?id=i4UfEAAAQBAJ>.
- BURKOV, Andriy. **The Hundred-Page Machine Learning Book**. 2019. ISBN 978-1999579500.
- CHEN, Chun-Teh; GU, Grace. Machine learning for composite materials. **MRS Communications**, v. 9, p. 1–11, mar. 2019. DOI: [10.1557/mrc.2019.32](https://doi.org/10.1557/mrc.2019.32).
- CHEN, Zhangxin. **Finite Element Methods and Their Applications**. Springer, 2005. (Scientific Computation).
- CHICCO, D.; WARRENS, M. J.; JURMAN, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. **PeerJ Comput Sci.**, 2021. DOI: [10.7717/peerj-cs.623](https://doi.org/10.7717/peerj-cs.623).
- CHOLLET, François. **Deep Learning with Python**. Manning, nov. 2017. ISBN 9781617294433.
- CHOLLET, François *et al.* **Keras**. 2015. Disponível em: <https://keras.io>.
- CONEJO, Luíza dos Santos. **Processamento e Caracterização de compósitos multifuncionais de resina furfurólica/CNT/Fibra de carbono**. 18 mar. 2019. Doutorado – Universidade Estadual Paulista (UNESP).
- FANG, Kai-Tai; LI, Runze; SUDJANTO, Agus. **Design and Modeling for Computer Experiments (Computer Science & Data Analysis)**. Chapman & Hall/CRC, 2005. ISBN 1584885467.
- FERREIRA, Wallace Gusmão. **Otimização eficiente global dirigida por metamodelos combinados: novos caminhos abertos pela aproximação por mínimos quadrados**. 2016.

Doutoramento – Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica, Campinas, SP.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. The MIT Press, 2016. 800 p. (Adaptive Computation and Machine Learning series). ISBN 978-0262035613.

GRECA, Ileana María; MOREIRA, Marco Antonio. Mental, physical, and mathematical models in the teaching and learning of physics. **Science Education**, v. 86, n. 1, p. 106–121, 2002. DOI: <https://doi.org/10.1002/sce.10013>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sce.10013>. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sce.10013>.

GUHA, R. V.; LENAT, Douglas B. CYC: A Mid-Term Report. **Applied Artificial Intelligence**, v. 5, n. 1, p. 45–86, 1991. DOI: [10.1080/08839519108927917](https://doi.org/10.1080/08839519108927917). Disponível em: <https://doi.org/10.1080/08839519108927917>.

HADDEN, C.M. *et al.* Mechanical properties of graphene nanoplatelet/carbon fiber/epoxy hybrid composites: Multiscale modeling and experiments. **Carbon**, v. 95, p. 100–112, 2015. ISSN 0008-6223. DOI: <https://doi.org/10.1016/j.carbon.2015.08.026>.

HINTON, Geoffrey E.; OSINDERO, Simon; TEH, Yee-Whye. A Fast Learning Algorithm for Deep Belief Nets. **Neural Comput.**, v. 18, n. 7, p. 1527–1554, jul. 2006. ISSN 0899-7667. DOI: [10.1162/neco.2006.18.7.1527](https://doi.org/10.1162/neco.2006.18.7.1527). Disponível em: <https://doi.org/10.1162/neco.2006.18.7.1527>.

HUGHES, Thomas Joseph Robert. **The finite element method: linear static and dynamic finite element analysis**. Courier Corporation, 2012.

KANIT, T. *et al.* Determination of the size of the representative volume element for random composites: statistical and numerical approach. **International Journal of Solids and Structures**, v. 40, n. 13, p. 3647–3679, 2003. ISSN 0020-7683. DOI: [https://doi.org/10.1016/S0020-7683\(03\)00143-4](https://doi.org/10.1016/S0020-7683(03)00143-4).

KAUSAR, Ayesha; RAFIQUE, Irum; MUHAMMAD, Bakhtiar. Aerospace Application of Polymer Nanocomposite with Carbon Nanotube, Graphite, Graphene Oxide, and Nanoclay. **Polymer-Plastics Technology and Engineering**, v. 56, n. 13, p. 1438–1456, 2017. DOI: [10.1080/03602559.2016.1276594](https://doi.org/10.1080/03602559.2016.1276594). Disponível em: <https://doi.org/10.1080/03602559.2016.1276594>.

KAW, Autar K. **Mechanics of Composite Materials**. 2. ed. USA: CRC Press, 2007. ISBN 0-8493-1343-0.

LORANDI, Natalia Pagnoncelli; CIOFFI, Maria Odilia Hilário; JUNIOR, Heitor Ornaghi. Análise Dinâmico-Mecânica de Materiais Compósitos Poliméricos. **Scientia Cum Industria**, v. 4, p. 48–60, 2016. DOI: <http://dx.doi.org/10.18226/23185279.v4iss1p48>. Disponível em: <https://core.ac.uk/download/pdf/205437478.pdf>.

MAHARAJ, Shiva; POLSON, Nick; TURK, Alex. Chess AI: Competing Paradigms for Machine Intelligence. **Entropy**, MDPI AG, v. 24, n. 4, p. 550, abr. 2022. DOI: [10.3390/e24040550](https://doi.org/10.3390/e24040550). Disponível em: <https://doi.org/10.3390/e24040550>.

MARLETT, Kristin. Hexcel 8552 AS4 Unidirectional Prepreg at 190 gsm & 35% RC Qualification Material Property Data Report. **CAM-RP-2010-002 Rev A**, 2011.

MARTÍN ABADI *et al.* **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. 2015. Software available from tensorflow.org. Disponível em: <https://www.tensorflow.org/>.

MATOS, M.A.S.; PINHO, S.T.; TAGARIELLI, V.L. Application of machine learning to predict the multiaxial strain-sensing response of CNT-polymer composites. **Carbon**, v. 146, p. 265–275, 2019. ISSN 0008-6223. DOI: <https://doi.org/10.1016/j.carbon.2019.02.001>.

MATOS, M.A.S.; PINHO, S.T.; TAGARIELLI, V.L. Predictions of the electrical conductivity of composites of polymers and carbon nanotubes by an artificial neural network. **Scripta Materialia**, v. 166, p. 117–121, 2019. ISSN 1359-6462. DOI: <https://doi.org/10.1016/j.scriptamat.2019.03.003>.

MCKINNEY, Wes. Data Structures for Statistical Computing in Python. *In*: WALT, Stéfan van der; MILLMAN, Jarrod (Ed.). **Proceedings of the 9th Python in Science Conference**. 2010. P. 56–61. DOI: [10.25080/Majora-92bf1922-00a](https://doi.org/10.25080/Majora-92bf1922-00a).

MENABREA, L. F. Sketch of The Analytical Engine. Invented by Charles Babbage. With notes upon the Memoir by the Translator Ada Augusta, Countess of Lovelace. **Bibliothèque Universelle de Genève**, n. 82, out. 1842.

MNIH, Volodymyr *et al.* **Playing Atari with Deep Reinforcement Learning**. 2013. Disponível em: <https://arxiv.org/pdf/1312.5602v1.pdf>. Acesso em: 7 abr. 2022.

Introduction to aerospace materials. *In*: MOURITZ, Adrian P. (Ed.). **Introduction to Aerospace Materials**. Woodhead Publishing, 2012. cap. 1, p. 1–14. ISBN 978-1-85573-946-8. DOI: <https://doi.org/10.1533/9780857095152.1>. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9781855739468500017>.

NETO, Flávio Levi; PARDINI, Luiz Carlos. **Compósitos estruturais: ciência e tecnologia**. 2. ed. São Paulo: Blucher, 2016.

OKADA, Shuntaro; OHZEKI, Masayuki; TAGUCHI, Shinichiro. Efficient partition of integer optimization problems with one-hot encoding. **Scientific reports**, v. 9, n. 1, p. 1–12, 2019.

OMAIREY, Sadik L; DUNNING, Peter D; SRIRAMULA, Srinivas. Development of an ABAQUS plugin tool for periodic RVE homogenisation. **Engineering with Computers**, Springer, v. 35, n. 2, p. 567–577, 2019.

PATHAN, M.V.; PATSIAS, S. *et al.* Measurements and predictions of the viscoelastic properties of a composite lamina and their sensitivity to temperature and frequency. **Composites Science and Technology**, v. 149, p. 207–219, 2017. ISSN 0266-3538. DOI:

<https://doi.org/10.1016/j.compscitech.2017.06.005>.

PATHAN, M.V.; PONNUSAMI, S.A. *et al.* Predictions of the mechanical properties of unidirectional fibre composites by supervised machine learning. **Scientific Reports**, v. 9, set. 2019. DOI:

[10.1038/s41598-019-50144-w](https://doi.org/10.1038/s41598-019-50144-w).

PATHAN, M.V.; TAGARIELLI, V.L. *et al.* A new algorithm to generate representative volume elements of composites with cylindrical or spherical fillers. **Composites Part B: Engineering**, v. 110, p. 267–278, 2017. ISSN 1359-8368. DOI:

<https://doi.org/10.1016/j.compositesb.2016.10.078>.

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PIETROCOLA, Maurício. A matemática como estruturante do conhecimento físico. **Caderno brasileiro de ensino de física**, v. 19, n. 1, p. 93–114, 2002.

QI, Zhenchao *et al.* Prediction of mechanical properties of carbon fiber based on cross-scale FEM and machine learning. **Composite Structures**, v. 212, p. 199–206, 2019. ISSN 0263-8223. DOI:

<https://doi.org/10.1016/j.compstruct.2019.01.042>.

REDDI, Sashank J; KALE, Satyen; KUMAR, Sanjiv. On the convergence of adam and beyond. **arXiv preprint arXiv:1904.09237**, 2019.

SCHRITTWIESER, Julian *et al.* Mastering Atari, Go, chess and shogi by planning with a learned model. **Nature**, n. 588, p. 604–609, 2020.

SHIRURU, Kuldeep. Neural Network Approach for Processing Substation Alarms. **International Journals of Power Electronics Controllers and Converters**, v. 1, p. 21–28, out. 2015.

SILVER, David *et al.* Mastering the game of Go with deep neural networks and tree search. **Nature**, n. 529, p. 484–489, 2016. DOI: [10.1038/nature16961](https://doi.org/10.1038/nature16961).

SIMPSON, T. W. *et al.* Metamodels for computer-based engineering design: Survey and recommendations. **Engineering with Computers**, v. 17, n. 2, p. 129–150, 2001. Copyright: Copyright 2017 Elsevier B.V., All rights reserved. ISSN 0177-0667. DOI: [10.1007/PL00007198](https://doi.org/10.1007/PL00007198).

SIMPSON, T.W. *et al.* Design and analysis of computer experiments in multidisciplinary design optimization: A review of how far we have come – Or not. English (US). *In: 12TH AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, MAO. American Institute of Aeronautics e Astronautics Inc., 2008. (12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, MAO)*. ISBN 9781563479472. DOI: [10.2514/6.2008-5802](https://doi.org/10.2514/6.2008-5802).

SOBESTER, A.; FORRESTER, A.; KEANE, A. **Engineering Design via Surrogate Modelling: A Practical Guide**. Wiley, 2008. ISBN 9780470770795. Disponível em: <https://books.google.com.br/books?id=uLMHmeMnRCcC>.

SONG, Wonsup; KRISHNASWAMY, Vikram; PUCHA, Raghuram V. Computational homogenization in RVE models with material periodic conditions for CNT polymer composites. **Composite Structures**, v. 137, p. 9–17, 2016. ISSN 0263-8223. DOI: <https://doi.org/10.1016/j.compstruct.2015.11.013>.

STROEVEN, Martijn; ASKES, Harm; SLUYS, L.J. Numerical determination of representative volumes for granular materials. **Computer Methods in Applied Mechanics and Engineering**, v. 193, p. 3221–3238, jul. 2004. DOI: [10.1016/j.cma.2003.09.023](https://doi.org/10.1016/j.cma.2003.09.023).

TURING, A. Computing machinery and intelligence. **Mind**, v. 59, p. 433–460, 1950.

UTZIG, Lukas *et al.* Modeling and simulation of the effective strength of hybrid polymer composites reinforced by carbon and steel fibers. **Journal of Materials Science**, v. 53, n. 1, p. 667–677, 2018. DOI: [10.1007/s10853-017-1512-9](https://doi.org/10.1007/s10853-017-1512-9).

WAN, Lei *et al.* Computational micromechanics-based prediction of the failure of unidirectional composite lamina subjected to transverse and in-plane shear stress states. **Journal of Composite Materials**, v. 54, n. 24, p. 3637–3654, 2020. DOI: [10.1177/0021998320918015](https://doi.org/10.1177/0021998320918015).

WOLLSCHLAGER, Jeff. **Material Characterization of Continuous Fiber Composites**. Altair. 25 mar. 2020. Disponível em: <https://web.altair.com/composites-webinar-4-on-demand?submissionGuid=bb42095a-af06-46be-a9ea-515d92eabf6a>.

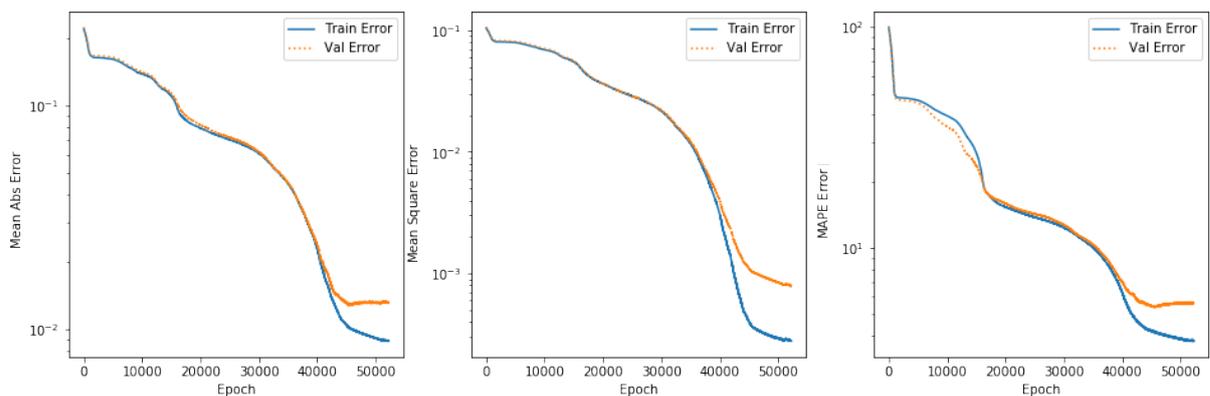
Apêndices

APÊNDICE A – Métricas em relação às épocas para todas as redes treinadas

Nessa seção são exibidos os gráficos relativos ao treinamento de todas as redes mostrada durante o desenvolvimento deste trabalho, mostrando a evolução dos erros nos dados de treino e de validação, para cada uma das redes.

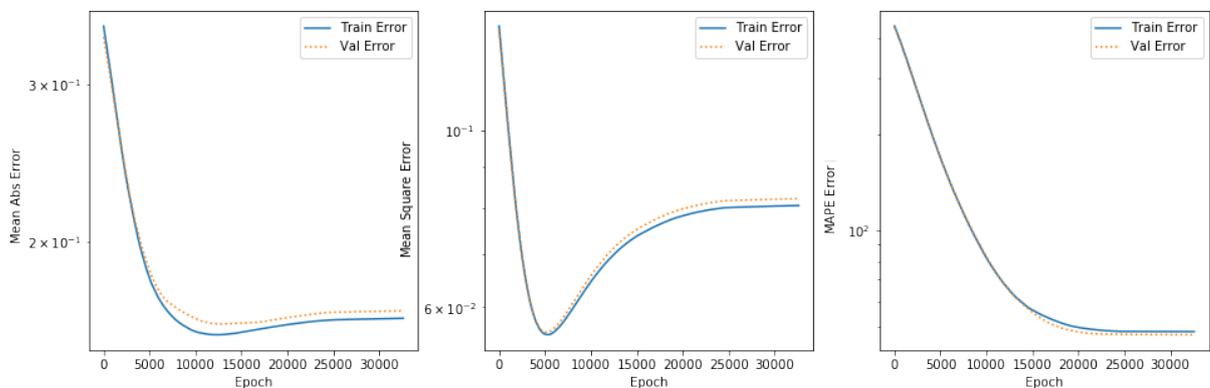
Os erros de treinamento e validação para os modelos iniciais, sem os dados categóricos podem ser encontrados nas Figuras 28 a 34.

Figura 28 – Erros de treinamento e validação ao longo das épocas para o Modelo 1 – 30 Camadas, 40 neurônios, reLU.



Fonte: Criada pelo autor.

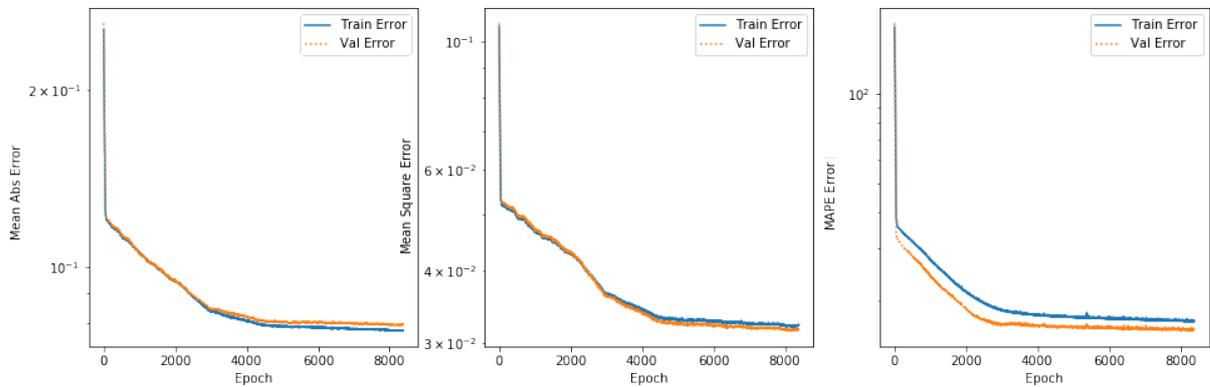
Figura 29 – Erros de treinamento e validação ao longo das épocas para o Modelo 2 – 30 Camadas, 40 neurônios, sigmoid.



Fonte: Criada pelo autor.

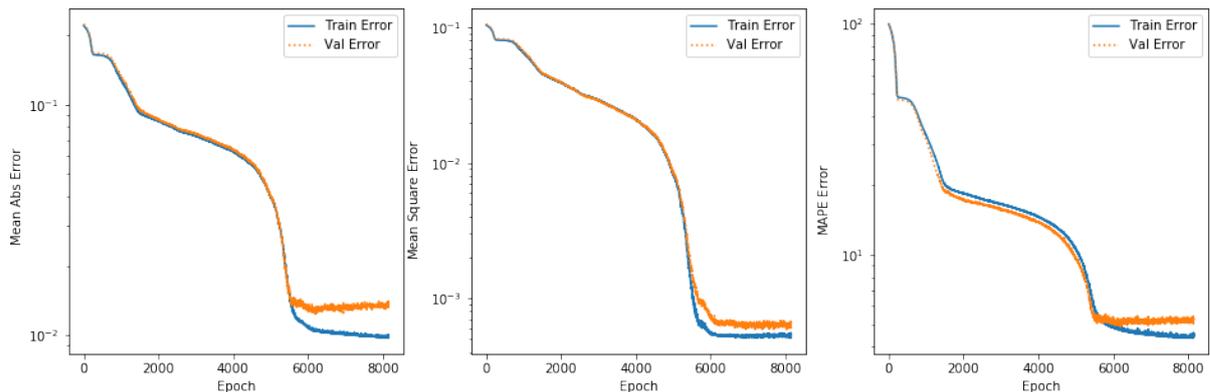
Os erros de treinamento e validação para os modelos finais, com os dados categóricos, podem ser encontrados nas Figuras 35, 36 e 37.

Figura 30 – Erros de treinamento e validação ao longo das épocas para o Modelo 3 – 30 Camadas, 40 neurônios, tanh.



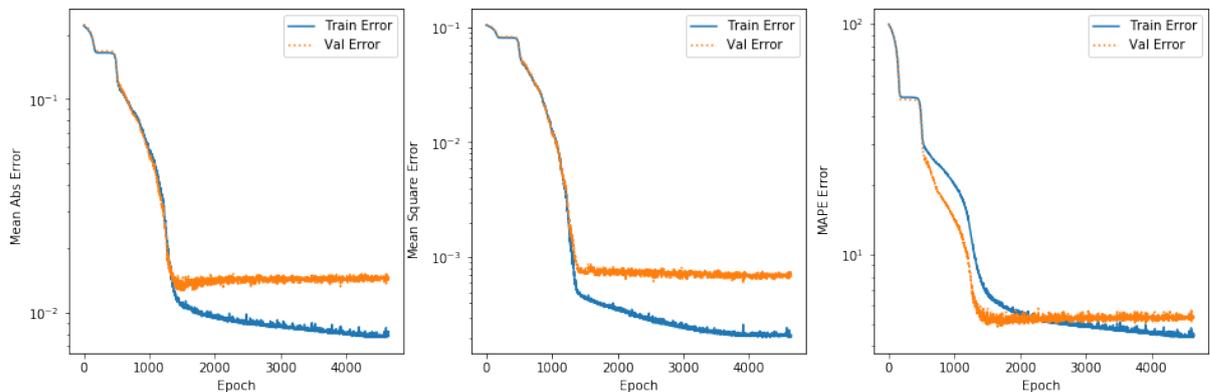
Fonte: Criada pelo autor.

Figura 31 – Erros de treinamento e validação ao longo das épocas para o Modelo 4 – 70 Camadas, 200 neurônios, reLU.



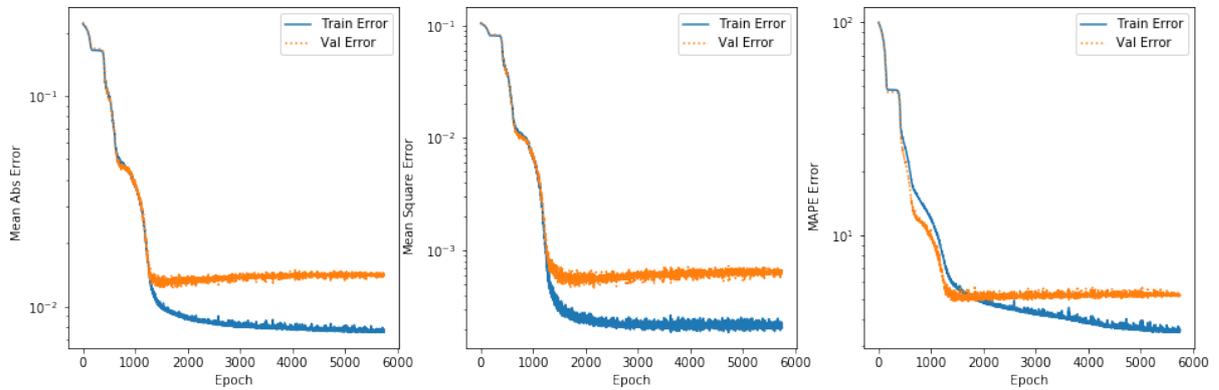
Fonte: Criada pelo autor.

Figura 32 – Erros de treinamento e validação ao longo das épocas para o Modelo 5 – 80 Camadas, 300 neurônios, reLU.



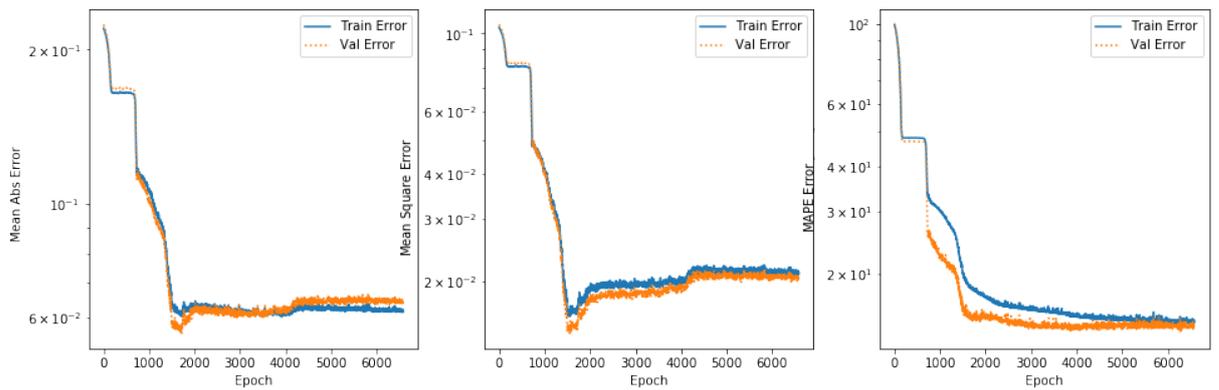
Fonte: Criada pelo autor.

Figura 33 – Erros de treinamento e validação ao longo das épocas para o Modelo 6 – 90 Camadas, 300 neurônios, reLU.



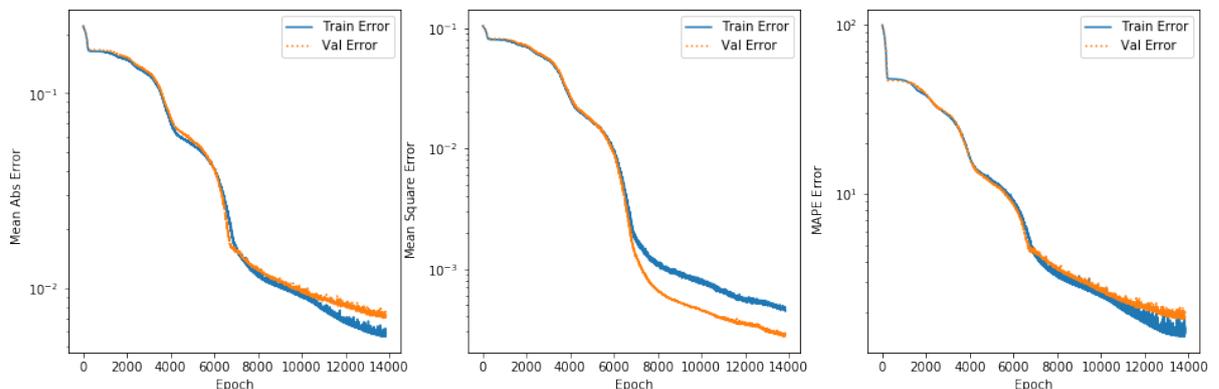
Fonte: Criada pelo autor.

Figura 34 – Erros de treinamento e validação ao longo das épocas para o Modelo 7 – 100 Camadas, 300 neurônios, reLU.



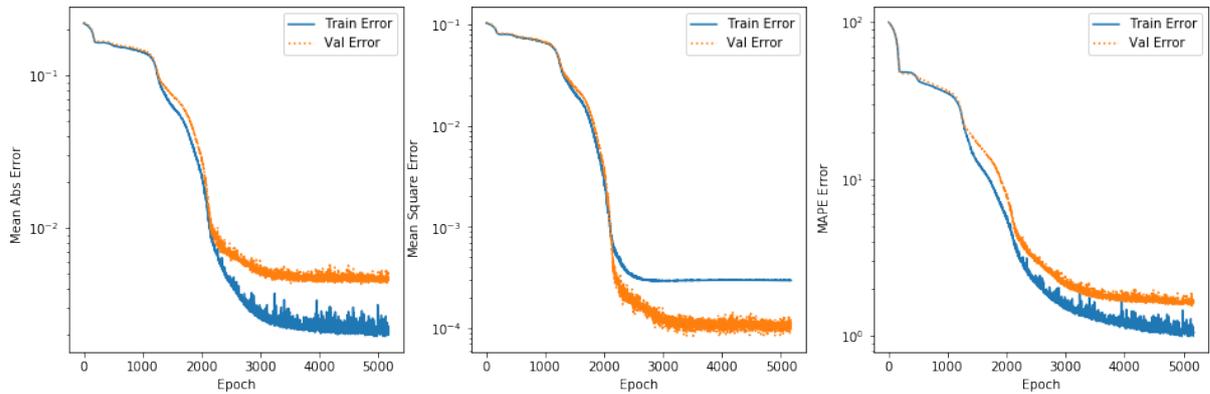
Fonte: Criada pelo autor.

Figura 35 – Erros de treinamento e validação ao longo das épocas para o Modelo 4 com dados categóricos — 70 Camadas, 200 neurônios, reLU.



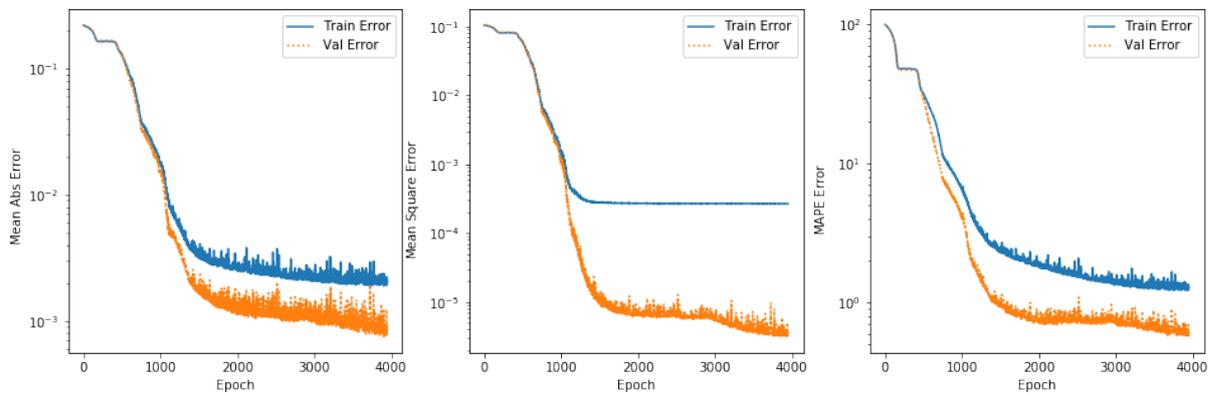
Fonte: Criada pelo autor.

Figura 36 – Erros de treinamento e validação ao longo das épocas para o Modelo 5 com dados categóricos — 80 Camadas, 300 neurônios, reLU.



Fonte: Criada pelo autor.

Figura 37 – Erros de treinamento e validação ao longo das épocas para o Modelo 6 com dados categóricos — 90 Camadas, 300 neurônios, reLU.



Fonte: Criada pelo autor.