

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ENGENHARIA ELÉTRICA
BACHARELADO EM ENGENHARIA ELETRÔNICA**

HILKIJA GAIUS TOSSO

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA MODELAGEM
VIRTUAL DE UM MOTOR DE COMBUSTÃO INTERNA CICLO OTTO
USANDO REDES NEURASIS**

TRABALHO DE CONCLUSÃO DE CURSO

**PONTA GROSSA
2020**

HILKIJA GAIUS TOSSO

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA MODELAGEM
VIRTUAL DE UM MOTOR DE COMBUSTÃO INTERNA CICLO OTTO
USANDO REDES NEURAIS**

Trabalho de Conclusão de Curso apresentado(a) como requisito parcial à obtenção do título de Bacharel em Engenharia Eletrônica, do Departamento Acadêmico de Engenharia Elétrica, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. MAX MAURO DIAS SANTOS

**PONTA GROSSA
2020**

TERMO DE APROVAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO - TCC

DESENVOLVIMENTO DE UMA FERRAMENTA PARA MODELAGEM VIRTUAL DE UM MOTOR DE COMBUSTÃO INTERNA CICLO OTTO USANDO REDES NEURAIS

Por

Hilkija Gaius Tosso

Monografia apresentada às 15 horas 30 min. do dia 11 de dezembro de 2020 como requisito parcial, para conclusão do Curso de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná, Câmpus Ponta Grossa. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação e conferidas, bem como achadas conforme, as alterações indicadas pela Banca Examinadora, o trabalho de conclusão de curso foi considerado APROVADO.

Banca examinadora:

Prof. Cristhiane Gonçalves	Me
Prof. Fernanda Cristina Côrrea	Me
Prof. Max Mauro Dias Santos	Orie
Prof. Josmar Ivanqui	Professor(a) re



Documento assinado eletronicamente por (Document electronically signed by) **MAX MAURO DIAS SANTOS, PROFESSOR DO MAGISTERIO SUPERIOR**, em (at) 11/12/2020, às 16:52, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por (Document electronically signed by) **FERNANDA CRISTINA CORREA, PROFESSOR DO MAGISTERIO SUPERIOR**, em (at) 11/12/2020, às 16:52, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por (Document electronically signed by) **CRISTHIANE GONCALVES, PROFESSOR DO MAGISTERIO SUPERIOR**, em (at) 11/12/2020, às 16:52, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por (Document electronically signed by) **JOSMAR IVANQUI, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em (at) 12/12/2020, às 08:11, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site (The authenticity of this document can be checked on the website) https://sei.utfpr.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador (informing the verification code) **1787030** e o código CRC (and the CRC code) **F6B61065**.

Dedico este trabalho a minha família e aos
meus amigos, pelos momentos de
ausência.

AGRADECIMENTOS

Este trabalho não poderia ser terminado sem a ajuda de diversas pessoas e/ou instituições às quais presto minha homenagem. Certamente esses parágrafos não irão atender a todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já peço desculpas àquelas que não estão presentes entre estas palavras, mas elas podem estar certas que fazem parte do meu pensamento e de minha gratidão.

A Deus, por tudo em minha vida.

Aos meus pais M. Etienne Tosso, Jeanne Kodjo e aos meus irmãos Perrerr Tosso, Amandine Tosso, Félicien Tosso, Rhoda Tosso, Gédéon Tosso e Noé Tosso pelo carinho, incentivo e total apoio em todos os momentos da minha vida.

A João Luiz Kovaleski, Nadia Véronique Jourda Kovaleski e aos seus filhos; a Eloiza Aparecida Silva Avila de Matos e à sua família; a Lorene Pecci e a todos os meus amigos pelo carinho e total apoio desde que eu cheguei no Brasil.

Ao meu orientador o Prof. Dr. Max Mauro Dias Santos, que me mostrou os caminhos a serem seguidos e pela confiança depositada.

A todos os professores e colegas do departamento, que ajudaram de forma direta e indireta na conclusão deste trabalho.

À Renault do Brasil e à Fundação Araucária, pela oportunidade de desenvolver esse projeto e pelo apoio financeiro.

Enfim, a todos os que de alguma forma contribuíram para a realização deste trabalho.

Faça o que você pode, com o que você
tem, no lugar onde você está.
(ROOSVERLT, Theodore, 1913).

RESUMO

TOSSO, Hilkija Gaïus. **Desenvolvimento de uma Ferramenta para Modelagem Virtual de um Motor de Combustão Interna Ciclo Otto usando Redes Neurais.** 2020. 64 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Eletrônica) – Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2020.

No presente trabalho, foi desenvolvido uma ferramenta de simulação numérica executável para modelar virtualmente alguns parâmetros de motores de combustão interna do ciclo Otto usando Redes Neurais Artificiais afim de auxiliar nos processos de calibração motor. Durante o processo de calibração existe uma certa quantidade de parâmetros a serem medidas e calibradas. O aumento desses parâmetros nos últimos anos tem aumentado a complexidade das tarefas a serem executadas, exigindo muito tempo e custos durante os testes com motor em bancada. A modelagem virtual do motor a combustão interna pode ser feita utilizando as técnicas de rede neural artificial por meio de alguns dados físicos sobre o motor e posteriormente podem ser usados durante o processo de calibração para avaliar os desempenhos dos motor em um espaço multi-dimensional sem necessidade testes adicionais. Para isso, foi proposto no presente trabalho uma metodologia que permite criar um modelo virtual do motor baseado em RNA. No entanto, existem várias arquiteturas possíveis de uma RNA e por isso, foi implementado um algoritmo genético para otimização da arquitetura da RNA, para que seja então possível configurar a melhor arquitetura de RNA para aquele motor específico a ser modelado. O método pode reduzir o tempo e custos em testes de bancada e, conseqüentemente, reduzir o tempo de desenvolvimento de calibração motor. Por fim, comprovamos a eficiência de nossa abordagem nos resultados experimentais de um projeto específico.

Palavras-chave: Motor de combustão interna. Modelagem. Simulação e Redes Neurais.

ABSTRACT

TOSSO, Hilkija Gaïus. **Development of a Tool for Virtual Modeling of an Internal Combustion Engine Otto Cycle Using Artificial Neural Networks**. 2020. 64 p. Final Coursework (Bachelor's Degree in Course Name) – Federal University of Technology – Paraná. Ponta Grossa, 2020.

The design, build and test of spark ignition engine has an important stage to be carry out on the test stands which allows to test performance, efficiency, components and calibration in the workflow of the engine development cycle. There are a certain level of variables to be measured and calibrated that increase the complexity of tasks to do, demanding for a high time and costs on bench testing. The virtual modeling of this complexity thermal machine might be done using the artificial neural network through instrumentation for acquire several physical variables and then evaluation several possible architectures. Basically, we capture a dataset that we presented to our network. Therefore, we developed a process that allows we identify and model the engine and put on in a virtual model based on ANN. However, there are a bunch of possible architectures available and through a genetic algorithm for optimization we can set up the best ANN architecture for that specific engine to be modeled. Our method can reduce the time and costs in bench testing taken to virtual environment and consequently reducing the time of development and time to market. Finally, we proof the efficiency of our approach in the experimental results of a specific project.

Keywords: Internal Combustion Engine. Modeling. Simulation and Neural Network.

LISTA DE ALGORITMOS

Algoritmo 1 – Algoritmo genético	42
--	----

LISTA DE ILUSTRAÇÕES

Figura 1 – Número de veículos produzidos por ano no Brasil entre 1957 e 2019	14
Figura 2 – Motor de Nicolaus Otto e Eugen Langen de 1867.	19
Figura 3 – Terminologia do motor.	20
Figura 4 – Os quatro tempos de um motor de combustão interna	21
Figura 5 – Development of sensors, actuators and electronic control systems for automobiles.	22
Figura 6 – Representação esquemática do sistema de controle EMS de um motor de combustão interna.	22
Figura 7 – As etapas para modelagem do motor usando métodos estatísticos.	24
Figura 8 – As conexões sinápticas entre neurônios.	26
Figura 9 – Tipos de função de ativação	27
Figura 10 – Tipos de arquitetura de Redes Neurais, (a) RNA <i>Feedforward</i> de uma camada, (b) RNA <i>Feedforward</i> com multi-camadas, (c) RNA <i>Feedback</i>	28
Figura 11 – As técnicas de modelagem de motor de combustão interna	29
Figura 12 – Combinação de um modelo caixa branca e modelo caixa preta	30
Figura 13 – Diagrama de fluxo de uma metodologia de calibração melhorada	31
Figura 14 – Método proposto para desenvolvimento do modelo caixa preta de motor a combustão interna.	35
Figura 15 – Sequencia bidimensional de 200, 500, 1.000 e 2.000 pontos gerada usando números pseudo-aleatórios.	36
Figura 16 – Sequencia bidimensional de 200, 500, 1.000 e 2.000 pontos gerada usando números quasi-aleatórios.	36
Figura 17 – Exemplo de genes de dois indivíduos e do descendente deles sem mutação de genes	41
Figura 18 – Ilustração da seleção por classificação	42
Figura 19 – Ilustração da interface gráfica da ferramenta desenvolvida para criação de modelos de RNAs	43
Figura 20 – Arquitetura da RNA para modelagem do motor	45
Figura 21 – Gráfico Box plot da evolução do <i>fitness</i> da população usando o algoritmo <i>Levenberg-Marquardt backpropagation</i>	47
Figura 22 – Histograma de erros e a regressão durante o treinamento da RNA usando o algoritmo <i>Levenberg-Marquardt backpropagation</i>	48
Figura 23 – Gráfico Box plot da evolução do <i>fitness</i> da população usando o algoritmo <i>Bayesian regularization backpropagation</i>	49
Figura 24 – Histograma de erros e a regressão durante o treinamento da RNA usando o algoritmo <i>Bayesian regularization backpropagation</i>	50
Figura 25 – FIM em função do ângulo VVT - 1700 rpm/540 mbar	53
Figura 26 – Sinal de erro na estimação feita pela RNA do FIM em função do ângulo VVT - 1700 rpm/540 mbar	53
Figura 27 – FIM em função da rotação em VVTIN 10 / VVTEX 20 / 540mbar	54
Figura 28 – Sinal de erro na estimação feita pela RNA do FIM em função da rotação em VVTIN 10 / VVTEX 20 / 540mbar	54

Figura 29 – FIM em função da pressão de admissão em 1700rrpm / VVTIN 20 / VVTEX 20	55
Figura 30 – Sinal de erro na estimação feita pela RNA FIM em função da pressão de admissão em 1700rrpm / VVTIN 20 / VVTEX 20	55

LISTA DE TABELAS

Tabela 1 – Parâmetros da arquitetura da RNA a serem otimizados	34
Tabela 2 – Equações das funções de ativação	34
Tabela 3 – Visão geral dos pontos de aquisição	44
Tabela 4 – Redução de 57% dos dados do motor	46
Tabela 5 – Melhor configuração encontrada para cada algoritmo de treinamento	47
Tabela 6 – Critérios de validação das Redes Neurais	49
Tabela 7 – Precisão da RNA treinada com 43% dos pontos usando o Levenberg-Marquardt backpropagation	49
Tabela 8 – Precisão da RNA treinada com 43% dos pontos usando a Bayesian regularization backpropagation	51
Tabela 9 – Precisão da RNA treinada com 65% dos pontos usando o Levenberg-Marquardt backpropagation	51
Tabela 10 – Precisão da RNA treinada com 65% dos pontos usando a Bayesian regularization backpropagation	52
Tabela 11 – Redução em 35% dos dados do motor	52

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

SIGLAS

ANFAVEA	Associação Nacional dos Fabricantes de Veículos Automotores
AFR	Razão ar/combustível
CA50	50% da massa combustível queimada
CO	Monóxido de carbono
CONAMA	CONSELHO NACIONAL DO MEIO AMBIENTE
DE	Differential evolution
ECU	Unidade de Controle do Motor
EMS	Sistema de Gerenciamento do Motor
FIM	Massa de injeção de combustível
GA	Algoritmo Genético
GNQA	Gerador de números quase-aleatórios
MAE	Erro Absoluto Médio
MAPE	Erro Percentual Absoluto Médio
MIMO-RNA	Rede Neural Artificial Multi-Input Multi-Output
MOR	Model order reduction
MPL	Rede neural perceptrons multicamadas
MVM	Modelos de valor médio
MSE	Erro Quadrático Médio
NA	Não se Aplica
NOx	Óxidos de nitrogênio
PROCONVE	Programa de Controle da Poluição do Ar por Veículos Automotores
PSO	Particle swarm optimisation
RBF	Função de base radial
RMSE	Raiz do Erro Quadrático Médio
RNA	Redes Neurais Artificiais
SVM	Máquinas de vetores de suporte
VVT	<i>Variable Valve Timing</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	MOTIVAÇÃO	16
1.2	OBJETIVO	16
1.2.0.1	Objetivo geral	16
1.2.0.2	Objetivos específicos	16
1.3	ORGANIZAÇÃO DO TRABALHO	17
2	REVISÃO DA LITERATURA	18
2.1	MOTORES DE COMBUSTÃO INTERNA	18
2.1.1	Definição	18
2.1.2	História e conceito	18
2.1.3	Controle motor	21
2.2	REDES NEURAIS ARTIFICIAIS	25
2.2.0.1	Definição	25
2.2.0.2	Conceito	25
2.2.0.3	Arquiteturas de Rede Neurais	25
2.2.0.4	Aprendizagem	26
2.3	MODELAGEM DE MOTOR A COMBUSTÃO INTERNA	28
2.3.1	Modelo caixa cinza	29
2.3.2	Modelo caixa preta	30
2.4	OTIMIZAÇÃO DA ARQUITETURA DE REDES NEURAIS ARTIFICIAIS	32
2.4.1	Algoritmos meta-heurísticos	32
2.4.2	Otimização da arquitetura de RNA usando algoritmos meta-heurísticos	33
3	MATERIAL E MÉTODOS	34
3.1	METODOLOGIA PROPOSTA PARA MODELAGEM DE MOTOR DE COMBUSTÃO INTERNA	34
3.1.1	Treinamento da Rede Neural usando um algoritmo de <i>backpropagation</i>	37
3.1.1.1	Levenberg-Marquardt <i>backpropagation</i>	37
3.1.1.2	Bayesian regularization <i>backpropagation</i>	39
3.1.2	Algoritmo Genético	40
3.2	INTERFACE GRÁFICA DA FERRAMENTA	42
4	RESULTADOS E DISCUSSÃO	44
5	CONCLUSÕES E PERSPECTIVAS	56
	REFERÊNCIAS	57

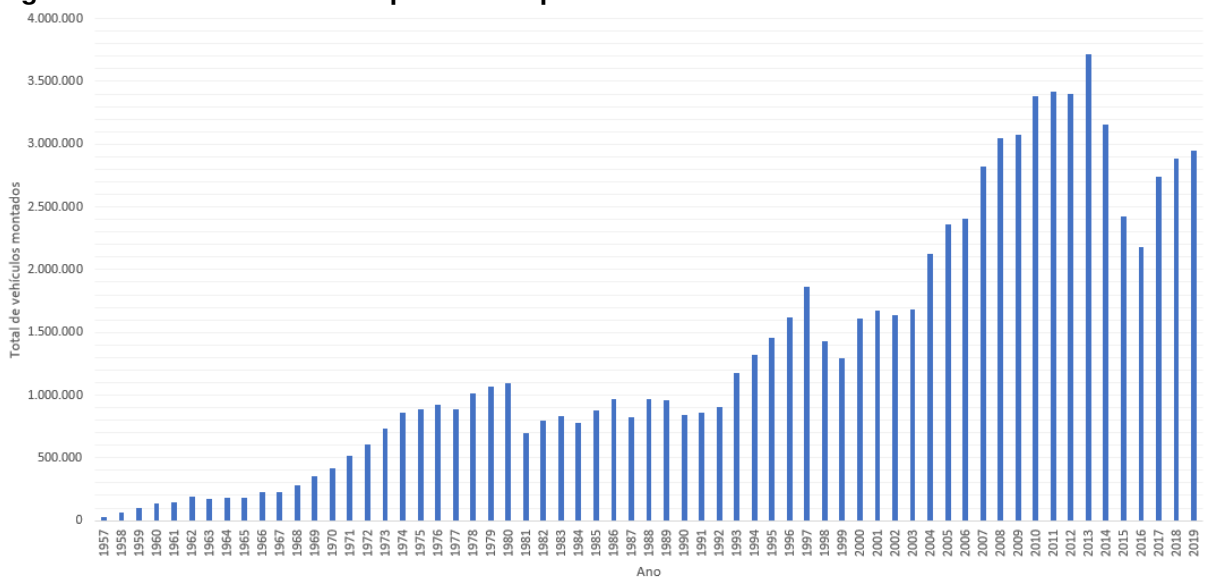
1 INTRODUÇÃO

Diante dos desafios do século, como os das mudanças climáticas e da escassez de recursos naturais, se torna cada vez mais urgente desenvolver novas tecnologias para reduzir a nossa pegada ecológica e ao mesmo tempo atender as necessidades da nossa civilização. O setor de transportes, especificamente o setor automotivo, é um dos setores que passou por profundas transformações nos últimos anos. Muitos países tem usado como arma a legislação para atacar o problema da insustentabilidade associada ao setor, tal como poluição do ar e sonora, congestionamento, acidentes, obesidade, etc (WHITMARSH; KÖHLER, 2010).

No Brasil por exemplo, o Programa de Controle da Poluição do Ar por Veículos Automotores (PROCONVE) cujo o objetivo é de reduzir a emissão de poluentes atmosféricos e de ruído de todos os veículos automotores vendidos no território nacional, tem sido um fator muito importante nos avanços tecnológicos observados na indústria automotiva nacional nos últimos anos (CARVALHO et al., 2013). Para continuar nesse sentido, O CONSELHO NACIONAL DO MEIO AMBIENTE (CONAMA) por meio da resolução N° 492, de 20 de dezembro de 2018, determinou novos limites máximos de emissão de poluentes para veículos rodoviários leves, de passageiros e comerciais que entrarão em vigor a partir do 1° de janeiro de 2022 (CONAMA, 2018).

Por outro lado, segundo os dados disponibilizado pela Associação Nacional dos Fabricantes de Veículos Automotores (ANFAVEA) como mostra a Figura 1 houve um forte crescimento no número de veículo produzidos no Brasil e um aumento na variedades de modelos disponíveis no mercado.

Figura 1 – Número de veículos produzidos por ano no Brasil entre 1957 e 2019



Fonte: Adaptado ANFAVEA (2020)

Para atender aos requisitos de regulação de emissões e desempenho, os mecanismos dos motores tornam-se a cada ano mais complexos como a *Dual VVT*, o levantamento de válvula variável e a injeção direta com múltiplos pulsos, aumentando assim a necessidade de maior tempo de desenvolvimento e de calibrações mais precisas do motor. Porém, essa necessidade de tempo de desenvolvimento é contra a demanda do mercado automotivo que exige novas tecnologias em menos tempo.

O processo de calibração do motor consiste em ajustar os parâmetros da Unidade de Controle do Motor (ECU) para melhorar a eficiência e o desempenho do motor, a fim de atingir objetivos como redução do consumo de combustível e redução das emissões de gases de escape para atender aos regulamentos de emissões locais, melhorando a potência do motor, durabilidade e dirigibilidade. Controlar o motor é uma das tarefas mais complexas para engenheiros de controle de sistemas e pesquisadores em todo o mundo por causa da grande quantidade de variáveis para controlar, como tempo de ignição por faísca, tempo de injeção de combustível, número de injeções de combustível, pressão de injeção de combustível, ajuste de VVT, razão ar/combustível (AFR), e também pelo fato dessas variáveis serem dependentes umas das outras, o que ocasiona um grande número de combinações (WANG et al., 2006).

O trabalho a seguir apresenta um estudo e desenvolvimento de uma ferramenta executável que modela alguns parâmetros necessários no processo de calibração de motores de combustão interna do ciclo Otto usando Redes Neurais Artificiais (RNA). Para isso, foram estudados dois algoritmos de treinamento de RNAs cujas arquiteturas foram otimizadas pelo Algoritmo Genético (GA), com o intuito de se encontrar uma arquitetura de rede ideal para cada parâmetro simulado na calibração dos motores. A ferramenta desenvolvida tem o intuito de otimizar o processo de calibração de motores tradicional, já que com a sua implementação, o tempo do processo de calibração pode ser reduzido consideravelmente.

O estudo de caso apresentado nesse trabalho se passa em uma montadora de veículos, localizada na cidade de São José dos Pinhais no estado de Paraná, que desenvolve motores de combustão interna do ciclo Otto. Nessa montadora, os projetos de motores passam pela fase de calibração de seus parâmetros antes de darem sequência à produção, neste período de calibração, existem ensaios físicos que precisam ser realizados com o motor em condições específicas, para que assim, possa ser feita a correta calibração dos parâmetros importantes para o seu funcionamento. Este cenário nos proporciona a possível aplicação de redes neurais artificiais na melhoria deste processo de calibração, otimizando o tempo dos profissionais envolvidos e aplicando conceitos de inteligência artificial na indústria automotiva.

1.1 MOTIVAÇÃO

O método de calibração usado atualmente na montadora começa pela campanha de calibração onde é feita um ensaio com motor em bancada para medir todo campo do motor. Em seguida, com base nos requisitos de cada projeto para que seja atendido as normas legislativas em vigor e as exigências do mercado, a equipe de calibração desenvolve a parametrização para calibrar o motor a ser usado para aquele projeto. Baseado nesses parâmetros de calibração é feita em ensaio de experimentos com motores em bancadas onde são medidas diversas variáveis físicas sobre a operação do motor para medir o seu desempenho e eficiência. Os dados resultantes desses ensaios, são então analisados pelo responsável de calibração são para ver se se os requisitos foram atendidos. Caso sim, a calibração segue para próxima etapa e caso contrário os parâmetros de calibração dão retocados e é feito um novo ensaio. Esse retrabalho representa um custo grande para a empresa, visto que os ensaios com motores em bancadas duram cerca de 15 horas.

Portanto, visto o poder computacional das redes neurais em resolver problemas complexos cujos a modelização matemática é inviável como no caso de alguns parâmetros no motor a combustão interna, a hipótese desse trabalho é modelar alguns parâmetros do motor usando RNA com base em dados adquiridos durante a campanha de calibração. Com esse modelo do motor, seria então possível simular virtual o desempenho e eficiência do motor antes de realizar os experimentos com motores em bancadas. Desta forma, os retrabalhos poderão ser reduzidos ou até mesmo evitados o que se traduzira por uma redução de custo para a empresa.

1.2 OBJETIVO

1.2.0.1 Objetivo geral

Desenvolver uma ferramenta de simulação numérica executável para modelar virtualmente alguns parâmetros de motores de combustão interna do ciclo Otto usando Redes Neurais Artificiais.

1.2.0.2 Objetivos específicos

- Investigar dois algoritmos de treinamento de redes neurais, (*Levenberg-Marquardt backpropagation and Bayesian regularization backpropagation*) disponibilizado pelo Matlab Neural Network Toolbox

- Implementar um método para encontrar de forma automática a arquitetura de redes neurais adequada para um dado problema de modelagem.
- Avaliar o funcionamento da ferramenta criando um modelo por meio dela.

1.3 ORGANIZAÇÃO DO TRABALHO

O restante do trabalho está estruturado da seguinte forma. O capítulo 2 cobre a revisão da literatura onde será introduzido os motores a combustão interna, redes neurais artificiais e apresentado alguns trabalhos relacionados a modelagem de motor a combustão interna e otimização da arquitetura de redes neurais artificiais. O capítulo 3 mostra o método proposto em detalhes e a avaliação do método proposto é discutido no capítulo 4. Finalmente, a conclusão e os trabalhos futuros são apresentados no capítulo 5.

2 REVISÃO DA LITERATURA

2.1 MOTORES DE COMBUSTÃO INTERNA

2.1.1 Definição

O motor de combustão interna é um motor térmico que converte energia química de um combustível em energia mecânica, geralmente disponibilizada em um eixo de saída rotativo (PULKRABEK, 2004).

Neste tipo de motor, a obtenção de trabalho é ocasionada pela sequência de processos realizados na mistura de ar e combustível na entrada do motor e nos produtos da combustão na saída (BRUNETTI, 2018).

2.1.2 História e conceito

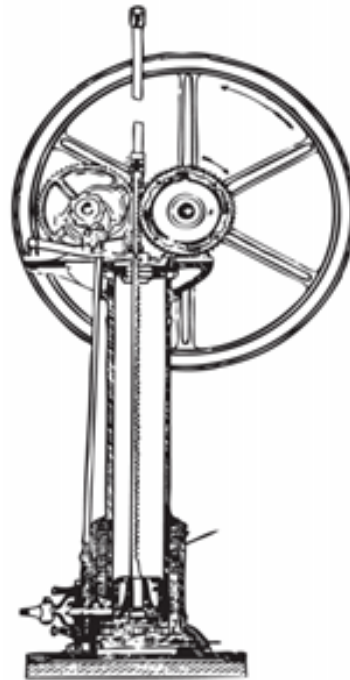
A idealização do motor de combustão interna se iniciou quando, em 1862, o físico francês Alphonse Beau de Rochas propôs as condições necessárias para que um motor de combustão interna, a quatro tempos, funcionasse com um máximo de economia (MILHOR, 2002). Entre estas condições definidas por Rochas, tem-se que a relação entre a superfície e o volume da câmara de combustão deve ser mínima, o processo de expansão deve ser realizado em velocidade máxima, a expansão deve ser máxima e no momento de iniciar-se a expansão, a pressão deve ser a máxima. Com base nas condições especificadas anteriormente, Rochas descreveu também o ciclo termodinâmico, que se comporta de forma essencialmente adiabática, por meio do qual a economia e a eficiência poderiam ser conseguidas, ciclo este que totalizava quatro tempos (FLÓREZ; AGRAMUNT, 2002).

O método de operação desejável para um motor de combustão interna também foi indicado por Rochas, seguindo as etapas de admissão, compressão, ignição e exaustão, que é basicamente o mesmo que ocorre nos motores de combustão interna até os dias atuais (TILLMANN, 2013).

A partir das definições já estabelecidas por Rochas, o engenheiro alemão Nikolaus August Otto e seu auxiliar mecânico Eugen Langen, desenvolveram em 1876 o primeiro motor de combustão interna de quatro tempos da história (RÍOS; GUZMÁN; SANTAMARÍA, 2002), conforme a Figura 2 a seguir.

A descoberta de um motor de combustão interna viável como o apresentado por Otto e Langen, possibilitou então o surgimento da indústria automotiva moderna, utilizando os mesmos princípios que são empregados até os dias atuais, embora sem-

Figura 2 – Motor de Nicolaus Otto e Eugen Langen de 1867.



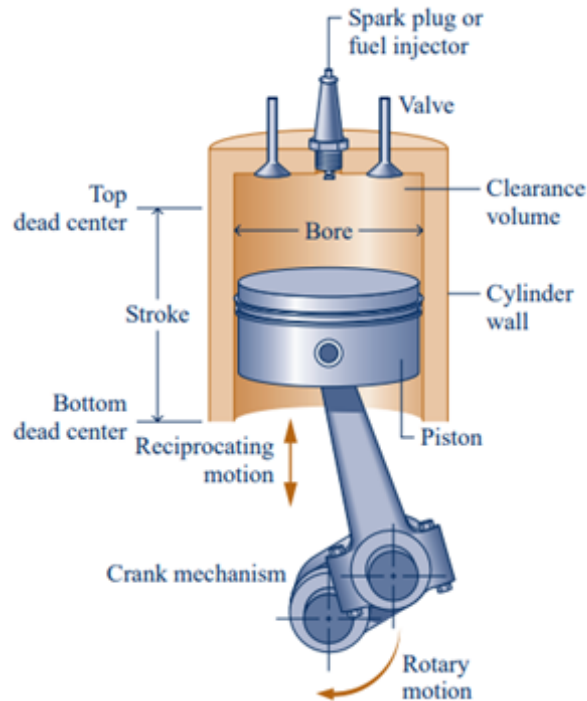
Fonte: Tillmann (2013).

pre venham a surgir novas alternativas para obtenção de trabalho mecânico facilitando a locomoção humana (FLÓREZ; AGRAMUNT, 2002). É válido também lembrar que os motores de combustão interna têm sido modificados e aprimorados ao longo dos anos, se ajustando as demandas provenientes da indústria automobilística. Desta maneira, os motores têm se tornado mais leves, econômicos e potentes, com a aplicação de novos materiais, melhorias em seus mecanismos, e outras alterações que resultaram em melhoria na eficiência volumétrica, assim como melhorias relacionadas à eficiência da combustão (MILHOR, 2002).

A Figura 3 representa um esboço de um motor de combustão interna que, segundo Shapiro et al (MORANM; HN et al., 2010), consiste em um pistão que se move dentro de um cilindro que contém duas válvulas. O esboço é rotulado com alguns termos especiais. O furo do cilindro é o seu diâmetro. O *stroke* é a distância a qual o pistão se move na direção vertical. Diz-se que o pistão está no ponto morto superior quando ele é movido para uma posição onde o volume do cilindro é mínimo. Este volume inferior é conhecido como volume de depuração. A taxa de compressão é definida como sendo o volume no ponto morto inferior dividido pelo volume no ponto morto superior. O movimento alternativo do pistão é convertido ao movimento rotativo por um mecanismo de manivela. Em um motor de combustão interna de quatro tempos, o pistão executa quatro cursos distintos dentro do cilindro para cada duas revoluções do Virabrequim.

Nos motores a quatro tempos, o ciclo de trabalho que se repete se dá pelas

Figura 3 – Terminologia do motor.



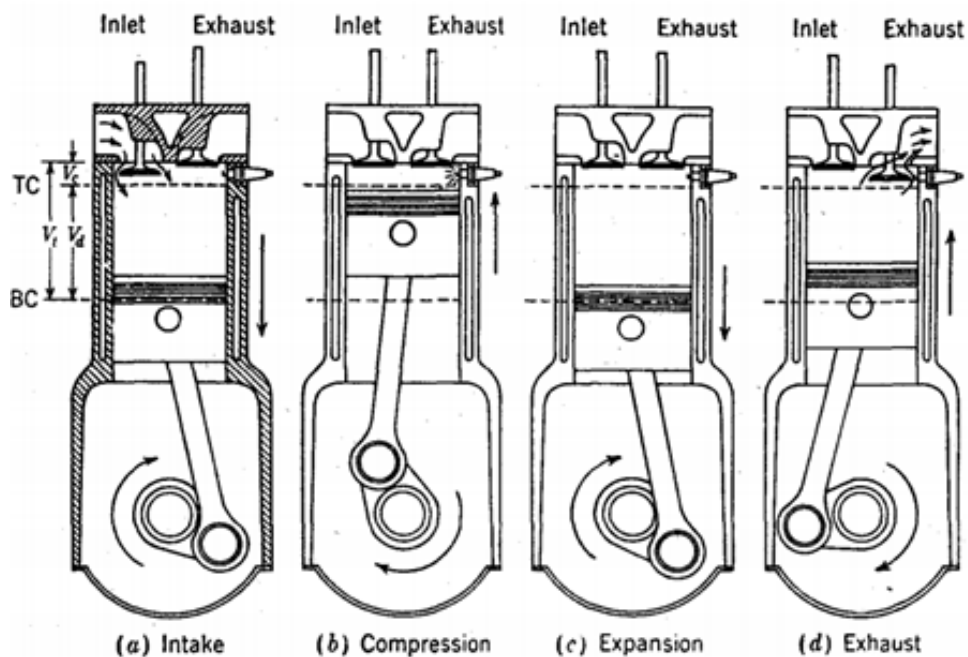
Fonte: MORANM, HN et al. (2010).

quatro etapas: admissão, compressão, expansão e exaustão. A figura 4 ilustra de maneira simplificada o comportamento desses quatro etapas que segundo o Heywood (HEYWOOD, 1988) são definidas da seguinte maneira:

1. Na primeira etapa, com a válvula de admissão aberta, o pistão faz um curso de admissão, do ponto TC ao BC, para puxar uma nova mistura de ar e combustível para o cilindro. Para aumentar a massa induzida, a válvula de admissão abre pouco antes do início do curso e fecha após o término.
2. A etapa de compressão ocorre quando ambas as válvulas estão fechadas e a mistura dentro do cilindro é comprimida a uma pequena fração de seu volume inicial. Se aproximando do fim desta, a combustão é iniciada e a pressão do cilindro sobe rapidamente.
3. O processo de expansão se inicia com o pistão em TC e termina em BC, onde os gases em alta temperatura e alta pressão empurram o pistão para baixo, forçando a manivela a girar. O trabalho realizado pelo pistão na etapa de expansão é cerca de cinco vezes maior do que o trabalho realizado durante a compressão. Conforme o pistão se aproxima de BC, a válvula de escape se abre dando início ao processo de exaustão deixando a pressão do cilindro próxima à pressão de exaustão.

4. Já na etapa de exaustão, os gases queimados restantes saem do cilindro pois a pressão deste pode ser substancialmente mais alta do que a pressão de exaustão. Portanto, o pistão se move em direção a TC, conforme os gases são varridos para fora do cilindro. Como o pistão da telha se aproxima de TC e a válvula de admissão se abre. Logo após o pistão chegar em TC, a válvula de escape se fecha e o ciclo se inicia novamente.

Figura 4 – Os quatro tempos de um motor de combustão interna .



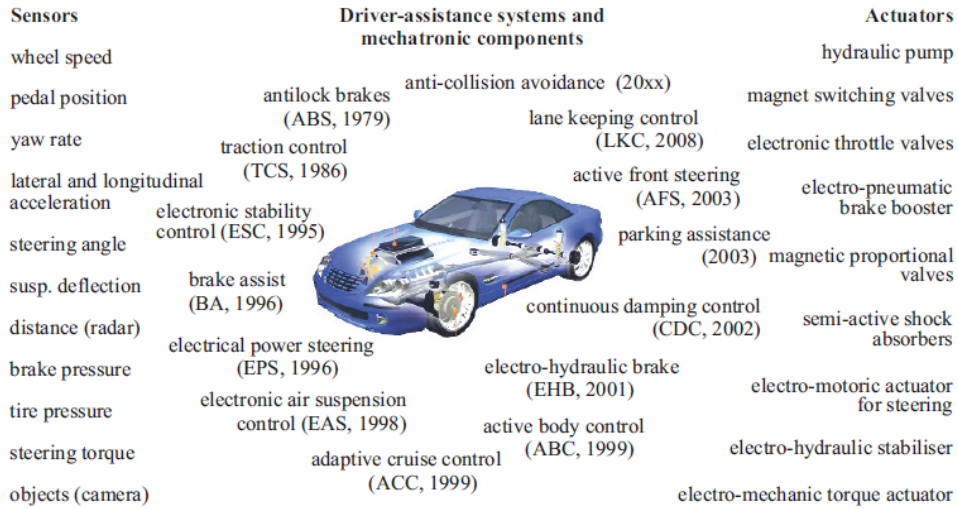
Fonte: Heywood (1988).

2.1.3 Controle motor

Os veículos com motor movido à combustão interna têm evoluído nos últimos anos em termo de performance, consumo de combustível e nível de emissões. Essa evolução se deu devido à substituição dos antigos sistemas de controle mecânico pelos sistemas de controle eletrônico que são capazes de fornecer as saídas desejadas para o motor (GONZALEZ; FLOREZ; ARAB, 2008). De fato, hoje em dia os motores de combustão interna possuem cerca de 15 à 25 sensores e 6 à 8 variáveis principais manipuladas que são controlados pela ECU (ISERMANN, 2014). A Figura 5 mostra a evolução de sensores, atuadores e controle eletrônico para veículos (ISERMANN, 2014).

Um Sistema de Gerenciamento do Motor (EMS) é constituído por vários sensores e atuadores. Os sensores são usados para monitorar as condições de operação em tempo real do motor enquanto os atuadores são usados para controlar por exemplo

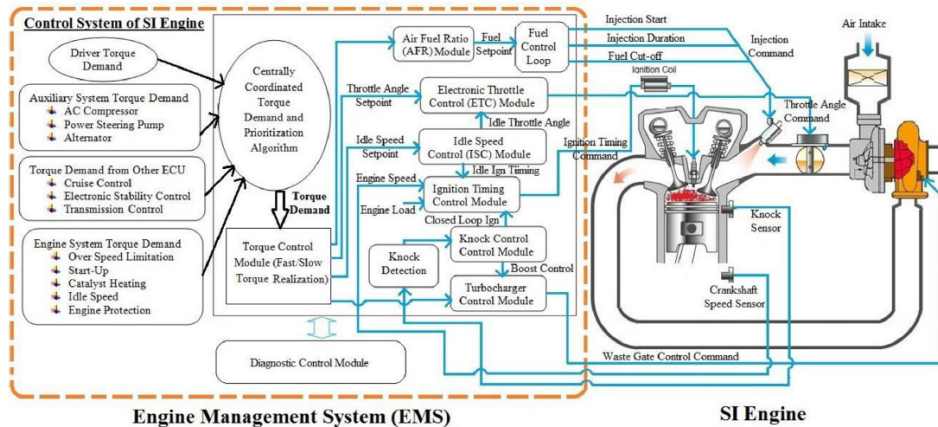
Figura 5 – Development of sensors, actuators and electronic control systems for automobiles.



Fonte: Isermann (2014).

o injetor, a vela de ignição, o acelerador, etc. O sinal de controle que é enviado para os atuadores é realizado pelo Sistema de controle EMS, que é composto por um grande número de módulos de controle dentro da sua arquitetura. A Figura 6 mostra uma representação da arquitetura do sistema de controle de um motor de combustão interna. Nessa Figura, pode-se ver alguns modelos como o *Air Fuel Ratio Module*, *Electronic Throttle Module*, *Torque Control Module*, *Idle speed Control Module*, etc. Todos esses módulos funcionam em paralelo à estrutura de controle de torque, no intuito de fornecer a potência desejada do motor conforme exigido pelo motorista (ASHOK; KUMAR, 2016).

Figura 6 – Representação esquemática do sistema de controle EMS de um motor de combustão interna.



Fonte: Ashok, Ashok e Kumar (2016)

Um EMS de motor possui diversas funções, com milhares de parâmetros, que são gerenciadas por algoritmo de controle de software no EMS e cada função precisa ser devidamente calibrada e testada junto ao software da unidade de controle

eletrônico (ECU) (ASHOK; ASHOK; KUMAR, 2016). Durante o desenvolvimento de sistemas de controle de motor automotivo, o projeto e a implementação dos algoritmos de controle é uma fase importantíssima, pois os motores possuem diferentes modos de operação, tais como o modo de inicialização, modo de marcha lenta, modo de operação normal, modo de saída de alta potência, etc (ASHOK; ASHOK; KUMAR, 2016). Então, ao ligar o motor, o EMS deve ser capaz de identificar de acordo com os dados coletados por diferentes sensores, as condições de operação do motor a fim de calcular e ajustar o tempo de injeção, avanço da ignição, ângulo do acelerador, etc (KIENCKE; NIELSEN, 2000; ISERMANN; HAFNER, 2001).

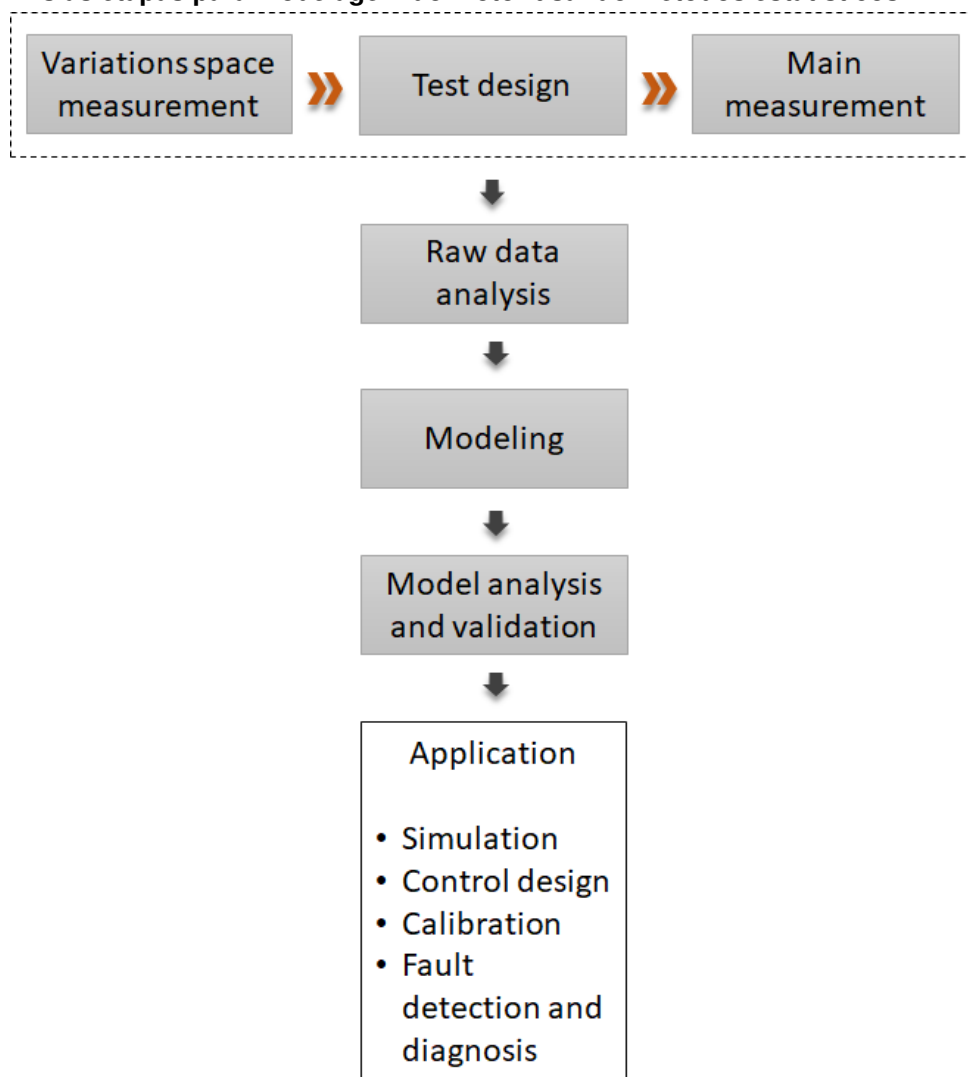
De acordo com (ASHOK; ASHOK; KUMAR, 2016), os algoritmos de controle do motor são desenvolvidos principalmente com base em modelos matemáticos e o controle não linear é implementado usando mapas do motor que foram derivadas de operações de testes com motor em bancada. Os processos de calibração são geralmente complicados e incluem diversos testes que cobrem vários parâmetros como o tempo de injeção de combustível e avanço de ignição, na forma de tabelas de pesquisa. Os principais testes são testes de calibração no estado estacionário, operação transiente, operação inativa. Todas essas tabelas de pesquisa fazem parte dos algoritmos de controle que são armazenados na memória do controlador. Além disso, o sistema de controle deve ser robusto o suficiente em relação às variações dos parâmetros que podem ocorrer no processo de produção bem como as variações de condições externas e envelhecimento.

Nos últimos anos, várias pesquisas têm sido realizadas com o objetivo de otimizar o processo de desenvolvimento de calibração usando um método baseado em modelo do motor (ROEPKE, 2014). Um método baseado em modelo de motor é uma combinação de princípios de modelagem baseados em equações e dados, usados para desenvolvimento de modelos dinâmicos em tempo real de alta precisão. Esses modelos são adequados para prever emissões de motor, desempenho e estados operacionais ao longo de ciclos operacionais altamente transitórios (ATKINSON; MOTT, 2005). A principal vantagem de se usar os métodos baseados em modelo de motor no processo de desenvolvimento de calibração é que eles permitem ao engenheiro de calibração avaliar os parâmetros em um espaço multidimensional sem a necessidade de testes adicionais.

A Figura 7 mostra por exemplo as etapas para modelagem do motor usando métodos estatísticos, com base em dados adquiridos em teste em bancadas, que consistem basicamente em adquirir e preparar os dados do motor, modelar o motor usando um método estatístico e finalmente, analisar e validar o modelo do motor. O modelo criado, pode ser então usado para simulações, controle, calibrações e para diagnósticos. Dentre os tipos de métodos usados, as Redes Neurais Artificiais (RNAs) têm demonstrado sua eficácia em diversos trabalhos (TURKSON et al., 2016; MOHA-

MED, 2019; ZHAO et al., 2012; ZHOU et al., 2008) devido ao seu poder de aproximação e capacidade de se adaptar a uma ampla variedade de não linearidades para baixo e alto número de parâmetros. No entanto, como a precisão de uma RNA está relacionada aos parâmetros ou hiper-parâmetros da sua arquitetura, então por um dado problema é necessário encontrar a arquitetura ideal da RNA (OJHA; ABRAHAM; SNÁŠEL, 2017). Portanto, alguns pesquisadores (BENARDOS; VOSNIAKOS, 2007; CARVALHO; RAMOS; CHAVES, 2011; ISLAM et al., 2014) propõem uma abordagem baseada em algoritmo meta-heurístico a fim de chegar a uma arquitetura de RNA otimizada.

Figura 7 – As as etapas para modelagem do motor usando métodos estatísticos.



Fonte: Adaptado de Isermann e Sequenz (2016)

2.2 REDES NEURAIS ARTIFICIAS

2.2.0.1 Definição

Uma rede neural artificial é um modelo computacional inspirado nos mecanismos biológicos que consistem em elementos de processamento, chamados neurônios, e conectados entre si com coeficientes, chamados pesos, ligados às conexões, que constituem a estrutura neuronal aos quais são anexados os algoritmos de treinamento e recuperação (KASABOV, 1996).

2.2.0.2 Conceito

As redes neurais artificiais simulam o mecanismo de aprendizado em organismos biológicos que contém células, chamadas neurônios (AGGARWAL et al., 2018). Os neurônios são conectados um ao outro com o uso de axônios e dendritos, e as regiões de conexão entre axônios e dendritos são chamadas de sinapses como mostrado na Figura 8a. Este mecanismo biológico é simulado em redes neurais artificiais, que contêm unidades de computação conhecidas como neurônios (AGGARWAL et al., 2018). Como mostra a Figura 8b, cada entrada para um neurônio é dimensionada com um peso, que afeta a função calculada nessa unidade.

Os neurônios são, portanto, funções não lineares dadas pela equação 1 a seguir onde φ , x_i e w_i são a função de ativação, os sinais de entrada s e os pesos sinápticos respectivamente. As funções de ativação tem como objetivo de limitar a amplitude nas saídas dos neurônios (HAYKIN, 2007).

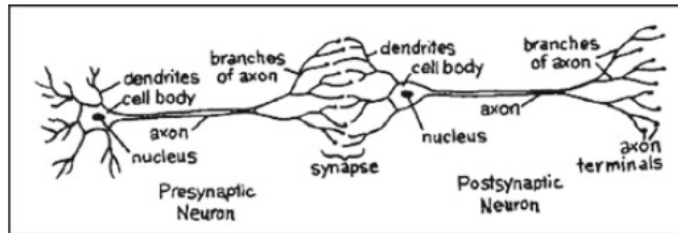
$$y_i = \varphi(w_0 + \sum_{i=1}^{n-1} w_i x_i) \quad (1)$$

A Figura 9 mostra um exemplo de três tipos básicos de funções de ativação que são: função de limiar, função linear e função sigmoide (HAYKIN, 2007). Dentre essas funções de ativação, as funções não-lineares são mais interessantes, pois, poucos são poucos os problemas que são linearmente separáveis (ZOU; HAN; SO, 2008).

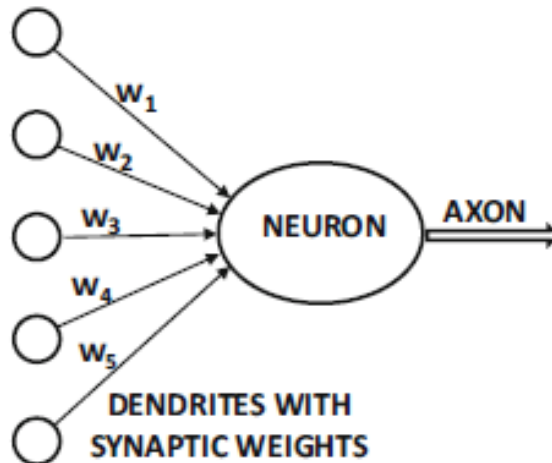
2.2.0.3 Arquiteturas de Rede Neurais

A arquitetura de uma RNA é definido como sendo a forma como os neurônios artificiais estão conectados e organizados entre si (ZOU; HAN; SO, 2008). De acordo com (HAYKIN, 2007), como mostrado na Figura 10, existem basicamente três classes de arquitetura de RNAs que são: Redes Neurais *Feedforward* de uma camada, redes neurais *Feedforward* com multi-camadas e as redes neurais Recorrente.

Figura 8 – As conexões sinápticas entre neurônios.



(a) Rede neural biológica



(b) Rede Neural Artificial

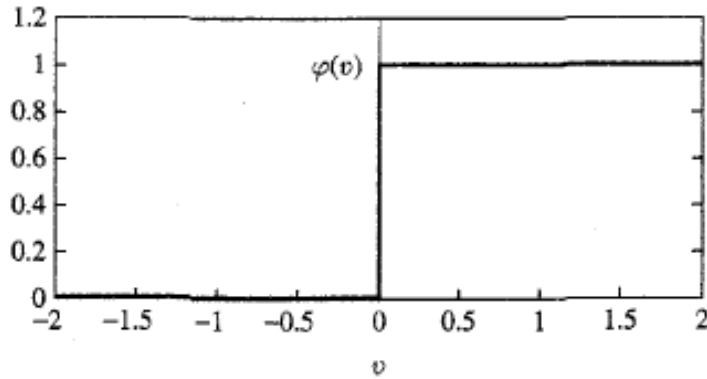
Fonte: Aggarwal et al. (2018)

Em uma Rede neural *Feedforward* de uma camada, representada pela Figura 10a, os neurônios são organizados em uma única camada saída onde são calculados as saídas dos neurônios de acordo com os dados que provem da camada de entrada. Essa arquitetura é chamado de *Feedforward*, pois, de fluxo das operações só vai em um sentido, da esquerda para direita. Por outro lado, em uma rede neural *Feedforward* com multi-camadas, representada pela Figura 10b, além das camadas de entrada e de saída, temos também uma ou mais camadas ocultas. Nesse tipo de arquitetura, o sinal de entrada que chega na camada de entrada passa pela rede, neurônios por neurônios da esquerda para a direita até produzir o sinal de saída. Por fim, como representada pela Figura 10c, em uma rede neural *Feedback* que pode ou não possui camadas ocultas, existe uma realimentação da camada de entrada pela camada de saída. Portanto, nessa arquitetura cada neurônio memoriza o seu valor anterior, pois, o seu valor futuro depende do sinal de entrada e do valor anterior.

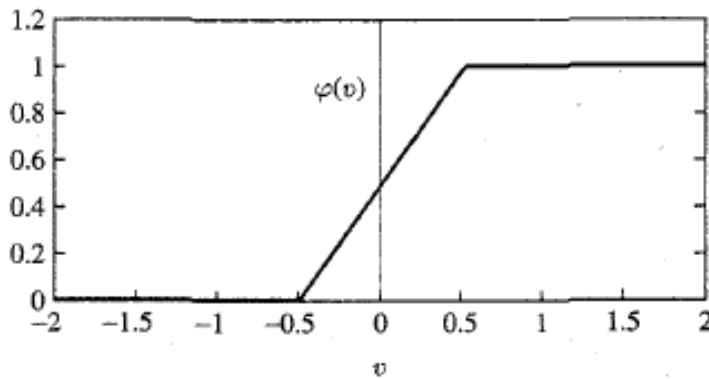
2.2.0.4 Aprendizagem

Uma vez escolhida a arquitetura da RNA, outra característica importante das RNAs é a aprendizagem. É durante essa fase que, por meio de algum algoritmo de aprendizagem, os pesos da RNA são ajustados até que os sinais de entrada da rede

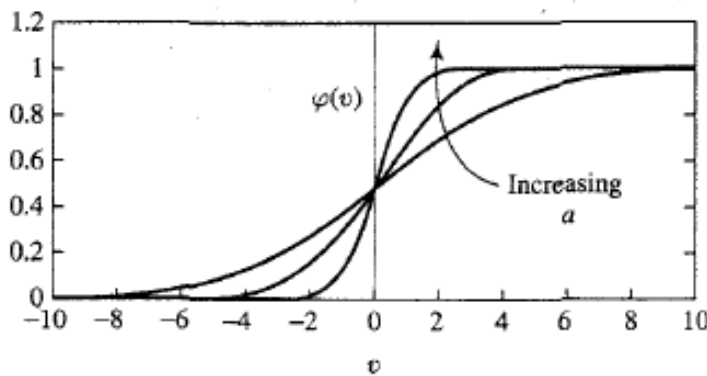
Figura 9 – Tipos de função de ativação



(a) função de limiar



(b) função linear



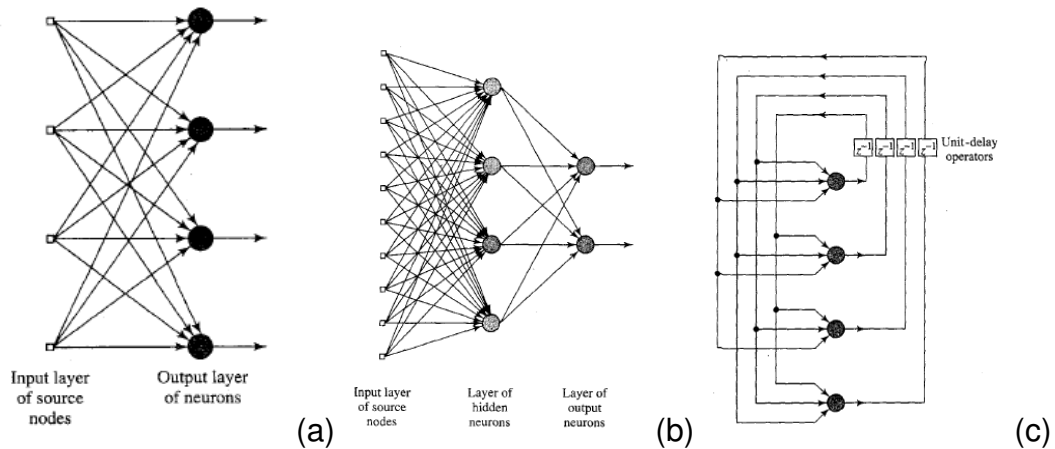
(c) função sigmoide

Fonte: Haykin (2007)

produzam um sinal de saída desejada. De acordo com (HAYKIN, 2007), a aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural são adaptados por meio de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira como ocorrem as alterações dos parâmetros.

Segundo (KASABOV, 1996), os algoritmos de aprendizagem são classificados em três grupos: a aprendizagem supervisionada, a aprendizagem não-supervisionada e a aprendizagem por reforço. Fala-se em aprendizagem supervisionada, quando dado um conjunto de sinais de entrada, são conhecidos os valores de sinais de saída desejados correspondentes. Então, durante esse processo de aprendizagem, são escolhidos dentro do conjunto de sinais de entrada um vetor de dado que alimenta a RNA

Figura 10 – Tipos de arquitetura de Redes Neurais, (a) RNA *Feedforward* de uma camada, (b) RNA *Feedforward* com multi-camadas, (c) RNA *Feedback*.



Fonte: Haykin (2007)

e então é calculado o vetor de saída produzido pela rede que é comparado com a saída desejada. Os pesos são então ajustados para que da próxima vez, as saídas produzidas pela RNA sejam o mais próximo possível da saída desejada. O processo de ajuste dos pesos é repetido até atingir a convergência. Já durante uma aprendizagem não-supervisionada, a RNA recebe apenas os sinais de entrada e deve então extrair uma característica interna ao conjunto. Por fim, durante uma aprendizagem por reforço, a RNA recebe os sinais de entrada e as saídas produzidas são julgadas para determinar se são boas ou não. Se a saída é considerada boa, então a RNA ganha uma recompensa e os pesos de conexão existentes são aumentados, caso contrário a RNA é punida e os pesos de conexão existentes são diminuídos.

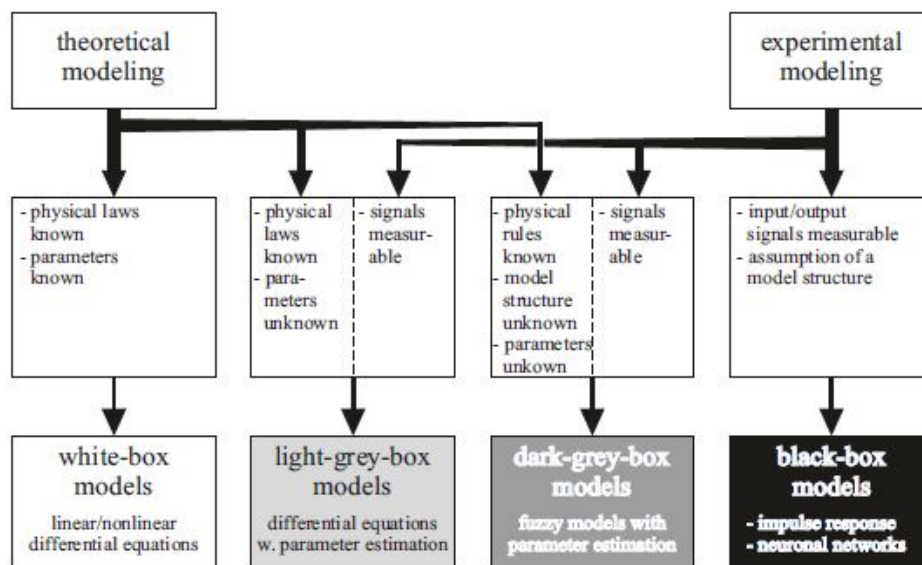
2.3 MODELAGEM DE MOTOR A COMBUSTÃO INTERNA

As técnicas de modelagem de motor podem ser divididas em duas categorias principais: a modelagem teórica e experimental também chamadas de modelos de caixa branca e caixa preta, respectivamente (ISERMANN, 2014). Modelos de caixa branca são adequados quando as leis e parâmetros físicos são conhecidos. Por exemplo, os modelos termodinâmicos, químicos, modelos de transferência de calor e modelos de dinâmica de fluidos, podem ser baseados em cálculo numérico que, por meio de equações de múltiplas dimensões, podem prever a dinâmica do motor como fluxo de massa de entrada, eficiência de combustão, liberação de calor do motor, geração de torque, consumo de combustível, emissão de poluentes (RAMADHAS; JAYARAJ; MURALEEDHARAN, 2006). A vantagem de se usar modelos caixas brancas é que eles são altamente confiáveis, porém, podem ser mais complexos e levam tempo para ser desenvolvidos.

Os modelos caixa-preta são baseados em medição para estabelecer, através

de um modelo de identificação, a relação entre o sinal de entrada e de saída medido. Apesar de terem um tempo de desenvolvimento mais curto comparando com os modelos caixa branca, a precisão desse tipo de modelo é dependente do tipo de método usado e da qualidade dos dados coletados. Por esse motivo, em muitas aplicações práticas (JUNG, 2020; ZANARDO et al., 2013; NICKMEHR, 2014), é feita a combinação dos modelos caixa branca e caixa preta, resultando em duas outras subcategorias de modelagem de motor que são os modelos de caixa cinza-claro e modelos de caixa cinza escuro como mostrado na Figura 11.

Figura 11 – As técnicas de modelagem de motor de combustão interna



Fonte: Adaptado de Isermann (2014)

2.3.1 Modelo caixa cinza

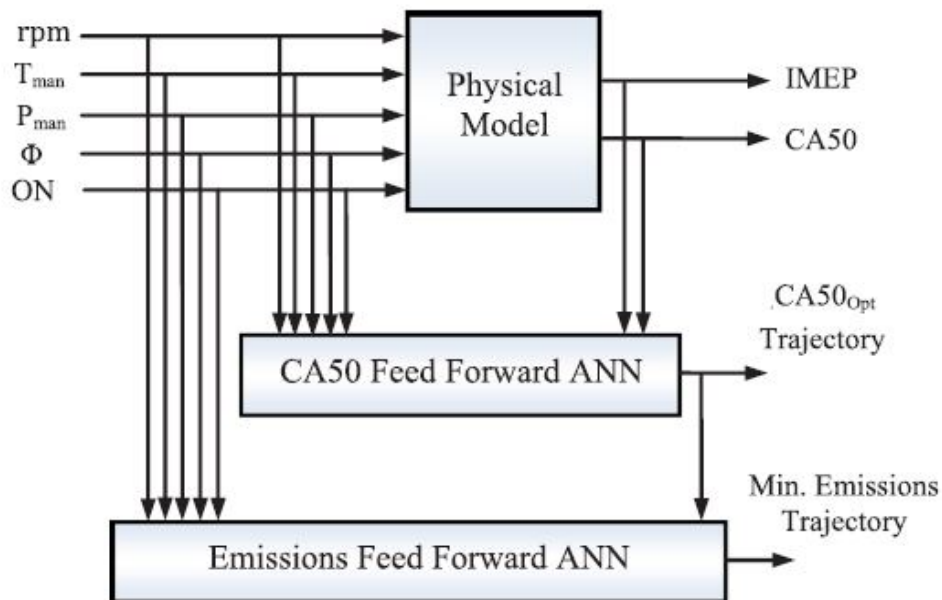
Modelos caixa cinza, são uma combinação dos modelos teóricos e experimentais. Esses modelos são adequados quando as leis físicas são conhecidos e a estrutura e/ou os parâmetros são desconhecidos Isermann (2014).

Em 2015, Shamekhi et Al (SHAMEKHI; SHAMEKHI, 2015) introduziram um novo método para melhorar os modelos de valor médio (MVM). O MVMs, são modelos caixa branca que fornecem uma relação matemática aproximada de um motor, descrevendo dinamicamente as suas variáveis física ao longo dos períodos de tempo (HENDRICKS, 1989). Para melhorar a precisão do MVM e deixá-o mais robusto, Shamekhi et Al propuseram um modelo de caixa cinza denominado Neuro-MVM que é uma combinação de Redes Neurais e do MVM. O modelo foi projetado para prever o torque, CO e NOx, recebendo como entradas a pressão e temperatura ambiente, o ângulo do acelerador, a razão ar/combustível, o ângulo da manivela de avanço da faísca,

a carga externa e a velocidade inicial. O estudo mostrou que não apenas o Neuro-MVM atinge grande precisão na previsão de seus modelos de saída, mas o modelo possui também mais confiabilidade do que o modelo baseado apenas em RNA.

Em outro trabalho conduzido por Nazoktabar et al (NAZOKTABAR et al., 2018) para desenvolver um modelo para motor de ignição por compressão de carga homogênea, eles usaram um modelo de caixa cinza que foi capaz de prever a pressão efetiva média indicada, o ângulo de manivela para CA50, monóxido de carbono e o hidrocarboneto total com os erros médios de 1,2 CAD, 0,4 bar, 10 PPM e 394 PPM, respectivamente. Conforme mostrado na Figura 12, este modelo consiste em dois modelos físicos e dois modelos de caixa preta baseada em RNA onde as saídas do modelo caixa branca são usadas como entrada média para os modelos de RNA.

Figura 12 – Combinação de um modelo caixa branca e modelo caixa preta



Fonte: Adaptado de Nazoktabar et al. (2018)

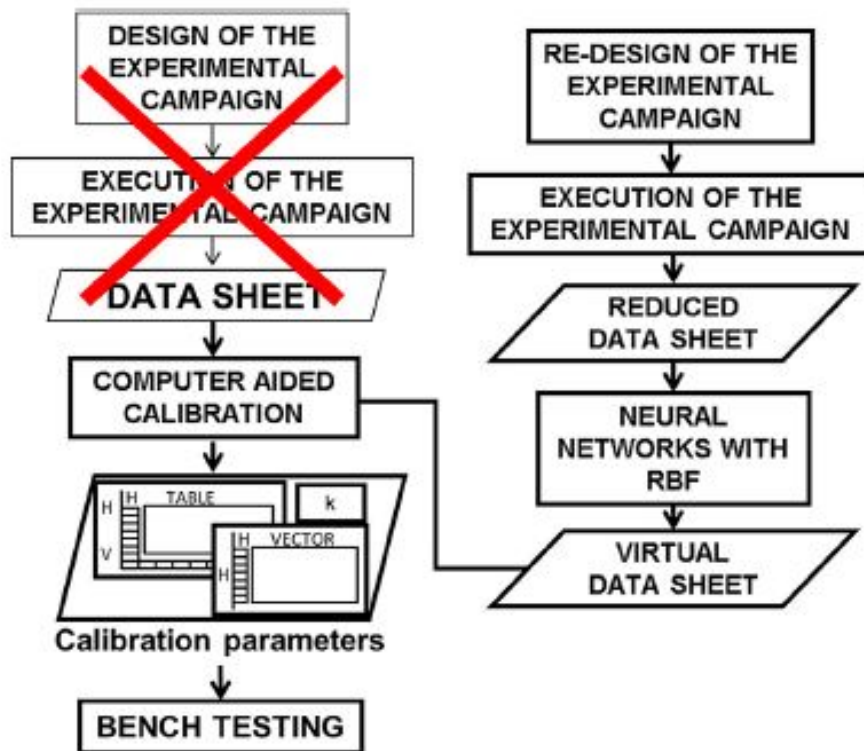
2.3.2 Modelo caixa preta

Apesar de serem menos confiáveis do que os modelos caixa branca, os modelos caixa-preta ainda são necessários, seja porque não é possível modelar teoricamente algumas partes do motor devido à falta de conhecimento sobre alguns parâmetros ou quando o tempo de construção do modelo físico não linear é incompatível com o tempo disponível para desenvolvimento do motor o que torna os modelos caixa branca inviável. Dada a importância dos modelos caixa-preta, a construção de modelos precisos despertou o interesse da comunidade de pesquisadores nos últimos anos.

Em 1996, um estudo conduzido por Nicolao et al (NICOLAO; SCATTOLINI; SIVIERO, 1996) investigou a eficácia do uso do modelo caixa preta para medir a eficiência volumétrica de um motor de combustão interna. Para isso, foi comparado o modelo aditivo, o modelo polinomial e dois modelos de Rede Neural (a função de base radial e os perceptrons multicamadas). O estudo mostrou que todos esses modelos de caixa preta podem ser usados com sucesso para descrever a eficiência volumétrica. No entanto, a rede neural de função de base radial (RBF) e o método de rede neural de perceptrons multicamadas (MLP) tiveram desempenho melhor do que os modelos aditivo e polinomial.

Em outro estudo conduzido por Francesco et Al em 2019 (NOLA et al., 2019) foi proposto, conforme mostrado na Figura 13, um método baseado em RNA para melhorar a metodologia de calibração através da redução da atividade de teste. A principal vantagem desse método é que ele necessita poucos dados para modelar virtualmente o motor usando uma rede neural RBF. Com o modelo criado, dados mais detalhados do motor virtual puderam ser gerados para calibração. A metodologia proposta foi testada estimando a eficiência volumétrica e mostrou que mesmo após uma redução de até 60 % dos dados experimentais normalmente adquiridos para fins de calibração, uma performance de calibração aceitável foi obtido.

Figura 13 – Diagrama de fluxo de uma metodologia de calibração melhorada



Fonte: Adaptado de Nola et al. (2019)

Nos últimos anos, além do método de modelagem de motor usando RNA, as

máquinas de vetores de suporte (SVM) tem sido investigadas para modelar o motor de combustão interna afim de prever o desempenho e as emissões do motor (NAJAFI et al., 2016; HAO et al., 2020). O algoritmo SVM é um método de aprendizado de máquina que foi introduzido em 1963 por Vladimir N. Vapnik e Alexey Ya. Chervonenkis. Os SVMs são usados como métodos de aprendizado supervisionado para regressão, classificação, detecção de valores discrepantes e, por esse motivo, podem ser usados para modelar relacionamentos complexos entre entrada e saída sem necessidade de entendimento físico direto do sistema. Em 2020, Norouzi et al (NOROUZI; ALI-RAMEZANI; KOCH, 2020) desenvolveram usando o algoritmo SVM um *Model order reduction* (MOR) para prever o estado estacionário da pressão efetiva média do freio e a emissão de NOx de um diesel. Eles compararam sua abordagem com uma rede neural *feedforward* de duas camadas de três neurônios na camada oculta, treinada com o algoritmo de treinamento Levenberg-Marquardt. Esta comparação mostrou que a abordagem SVM foi mais precisa e rápida do que a abordagem ANN. No entanto, é importante notar que a arquitetura da RNA usada para esta comparação com seus respectivos número de camadas e neurônios foi escolhida com base em outro estudo semelhante (HE; RUTLAND, 2004) (NIU et al., 2017).

2.4 OTIMIZAÇÃO DA ARQUITETURA DE REDES NEURAIS ARTIFICIAIS

2.4.1 Algoritmos meta-heurísticos

Os algoritmos meta-heurísticos são algoritmos geralmente inspirados da natureza e usados para resolução de problemas de otimização. Esses algoritmos são principalmente caracterizados pela seleção da melhor solução que assegura a convergência para a solução ótima e a randomização que assegura a diversidade afim de evitar mínimos locais (YANG, 2010a).

Embora os algoritmos meta-heurísticos possam ser usados para resolver vários problemas de otimização, há uma grande variedade deles e a performance de cada um desses algoritmo pode variar de acordo com o problema a ser resolvido. Em estudo publicado em 2020 por Absalom et al (EZUGWU et al., 2020), foi analisado com base no mesmo conjunto de problemas de benchmark, o desempenho de doze algoritmos meta-heurísticos que são o algoritmo genético (GA), *particle swarm optimisation* (PSO), *ant colony optimisation* (ACO), *symbiotic organisms search* (SOS), *cuckoo search* (CS), *firefly algorithms* (FA), *artificial bee colony* (ABC), *bat algorithms* (BA), *differential evolution* (DE), *flower pollination algorithms* (FPA), *invasive weed optimisation* (IWO) e o *BeeA* (HOLLAND, 1992; WANG et al., 2003; DORIGO; BIRATTARI; STUTZLE, 2006; CHENG; PRAYOGO, 2014; YANG; DEB, 2009; FISTER et al., 2013;

KARABOGA; BASTURK, 2007; YANG, 2010b; WAHAB; NEFTI-MEZIANI; ATYABI, 2015; YANG, 2012; MEHRABIAN; LUCAS, 2006; PHAM; CASTELLANI, 2009). Entre esses algoritmos, os que tiveram melhor desempenho em todos os problemas de teste são DE, PSO e GA.

2.4.2 Otimização da arquitetura de RNA usando algoritmos meta-heurísticos

As RNAs são geralmente definidas por seu tipo de neurônio, arquitetura, algoritmo de aprendizagem (SUBANA; SAMARASINGHE, 2016). Dependendo do problema a ser resolvido, é necessário definir o número adequado de camadas ocultas e neurônios em cada camada, pois esses parâmetros afetam a aprendizagem e a capacidade de generalização da RNA. No entanto, não há um método teórico para encontrar esses parâmetros. Para contornar esse problema, os algoritmos meta-heurísticos podem ser usados para otimizar as configurações da arquitetura de RNAs.

Em 2011, Carvalho et al (CARVALHO; RAMOS; CHAVES, 2011) investigaram quatro algoritmos de otimização (*Generalized Extremal Optimization, Variable Neighborhood Search, Simulated Annealing e GA*) a fim de propor um novo método adaptável para otimizar uma arquitetura de rede neural *feedforward*. Os parâmetros de arquitetura da RNA que foram otimizados foram: a taxa de aprendizagem, o termo momentum, a função de ativação onde foram investigado o Tanh, o sigmóide, o log e o gaussiana, o número de camadas ocultas em um intervalo de 1 a 3 e o número de neurônios em cada camada oculta em um intervalo de 1 a 32. O estudo foi conduzido com base em conjuntos de dados de benchmark padrão do repositório de aprendizado de máquina da Universidade da Califórnia em Irvine e mostrou que a *Variable Neighborhood Search* e o GA tiveram melhor performance do que os demais. Apesar do custo computacional, esta metodologia mostrou-se mais eficiente do que o método convencional de tentativa e erro para projetar uma arquitetura de RNA.

A próxima seção apresenta a metodologia proposta nesse trabalho para modelagem de Motor de Combustão Interna usando uma RNA *Feedforward* cujo a arquitetura será otimizada usando o GA.

3 MATERIAL E MÉTODOS

3.1 METODOLOGIA PROPOSTA PARA MODELAGEM DE MOTOR DE COMBUSTÃO INTERNA

A metodologia empregada neste trabalho está descrita na Figura 14 e tem como objetivo implementar um método adaptável para melhorar a performance de um modelo caixa preta para modelagem de motor de combustão interna baseada em uma Rede Neural Artificial *Multi-Input Multi-Output* (MIMO-RNA) afim de de auxiliar no processo de desenvolvimento de calibração dentro de uma montadora de veículos situada em São José dos Pinhais(PR) . Para isso, definiu-se primeiramente os parâmetros da arquitetura do MIMO-RNA a serem otimizados durante este processo. Neste trabalho, considerou-se como parâmetros, conforme mostra a Tabela 1, a função de transferência, o número da camada, o número de neurônios em cada camada e o número de épocas que representam o número de interações feitas pelo algoritmo de aprendizagem para ajustar os pesos da RNA. A Tabela 2 mostra as equações da função de ativação investigada neste trabalho.

Tabela 1 – Parâmetros da arquitetura da RNA a serem otimizados

Architecture's Parameter	Values
Hidden layers	{1, 2, 3, 4, 5}
Neurones in the first layer	{1, 2, 3, ..., 40}
Neurones in the second layer	{1, 2, 3, ..., 40}
Neurones in the third layer	{1, 2, 3, ..., 40}
Neurones in the fourth layer	{1, 2, 3, ..., 40}
Neurones in the fifth layer	{1, 2, 3, ..., 40}
Epoch	{1, 2, 3, ..., 1000}
Activation Function in each layer	{elliotsig, logsig, radbas, tansig}

Fonte: Autoria própria

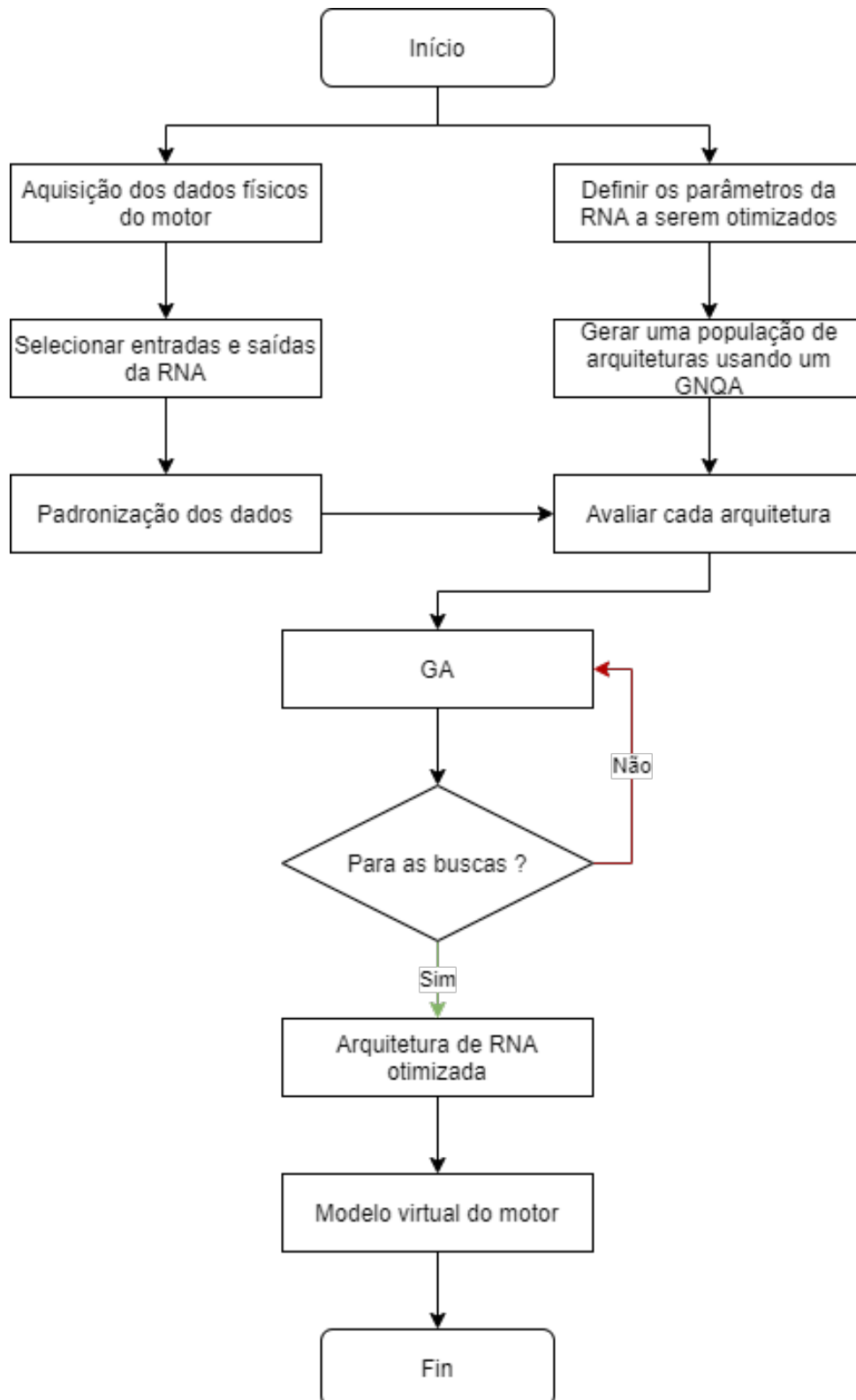
Tabela 2 – Equações das funções de ativação

Activation Function	Equation
Elliot symmetric sigmoid	$f(u) = \frac{u}{1+ u }$
Log-Sigmoid	$f(u) = \frac{u}{1+e^{-u}}$
Radial basis	$f(u) = e^{-u^2}$
Tan-Sigmoid	$f(u) = \frac{2}{1+e^{-2u}} - 1$

Fonte: Autoria própria

Inicialmente, gerou-se um conjunto de arquiteturas de MIMO-RNA *feedforward* (população inicial) usando um gerador de números quase-aleatórios (GNQA) para que tenha-se uma distribuição uniforme das arquiteturas dentro do espaço de busca

Figura 14 – Método proposto para desenvolvimento do modelo caixa preta de motor a combustão interna.

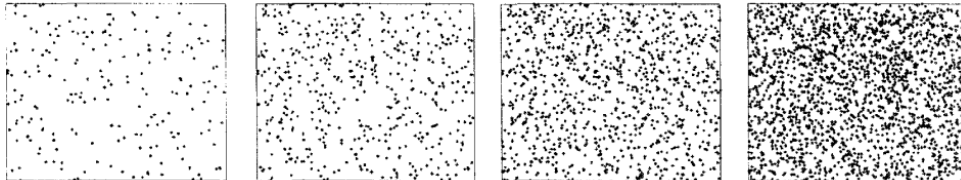


Fonte: Autoria própria

definido na Tabela 1. As Figuras 15 e 16 mostram um exemplo de sequenciais bi-dimensionais de 200, 500, 1.000 e 2.000 pontos gerada usando números pseudo-

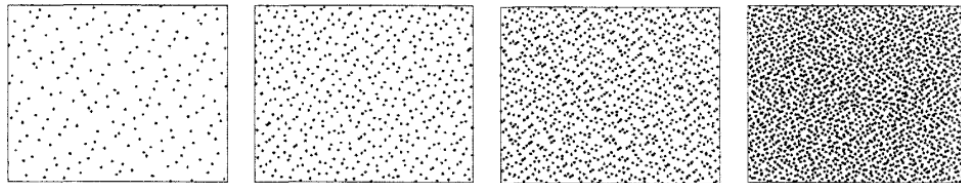
aleatórios e quase-aleatórios. O uso de GNQA foi motivado pela propriedade que esses geradores possuem de minimizar a discrepância entre a distribuição dos pontos gerados, possibilitando que eles apresentem melhores resultados na resolução de problemas de otimizações usando o GA, comparado à sequencias gerados pseudo-aleatoriamente (MAARANEN; MIETTINEN; MÄKELÄ, 2004).

Figura 15 – Sequencia bidimensional de 200, 500, 1.000 e 2.000 pontos gerada usando números pseudo-aleatórios.



Fonte: Maaranen, Miettinen e Mäkelä (2004)

Figura 16 – Sequencia bidimensional de 200, 500, 1.000 e 2.000 pontos gerada usando números quasi-aleatórios.



Fonte: Maaranen, Miettinen e Mäkelä (2004)

Em seguida, para avaliar as arquiteturas da MIMO-RNA que foram, os sinais de entrada e saída do modelo do motor são adquiridos em ensaios em bancadas de teste. Esses dados são discretizados e então são selecionados, de acordo com a calibração a ser realizada, os dados de entrada e saída da MIMO-RNA. Esses dados estão representados por uma matriz $X[i,j]$ e como os sinais podem ter diferentes ordens de magnitude, a matriz $X[i,j]$ é padronizados em um intervalo $[a, b]$ para minimizar o erro durante o treinamento da MIMO-RNA usando a equação 2.

$$X_{i,j} = \frac{(a - b) * X_{i,j} - \text{minimo}(X_j)}{\text{maximo}(X_j) - \text{minimo}(X_j) + b} \quad (2)$$

Os dados sendo padronizados, as MIMO-RNA são alimentadas com os sinais de entrada para serem treinadas com um algoritmo de aprendizagem supervisionada e em seguida suas performances são avaliadas. A performance das RNAs são geralmente avaliadas usando métricas de erro. A métricas mais comum são o Erro Absoluto Médio (MAE), o Erro Quadrático Médio (MSE), o Raíz do Erro Quadrático Médio (RMSE) e o Erro Percentual Absoluto Médio (MAPE) dadas pelas equações 3, 4, 5 e 6 respectivamente onde n representa o número de dados, y o valor efetivo e t o valor estimados pela RNA (SIQUEIRA et al., 2018).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - t_i| \quad (3)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - t_i)^2 \quad (4)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - t_i)^2} \quad (5)$$

$$MAPE = \frac{100}{n} \sum_{i=1}^n \frac{|y_i - t_i|}{y_i} \quad (6)$$

Após avaliação da população inicial, a próxima geração de arquiteturas é gerada pelo algoritmo genético até encontrar a arquitetura ótima.

Neste trabalho, foi investigado dois algoritmos de aprendizagem supervisionada disponível dentro do MATLAB *Neural Network Toolbok* (BEALE; HAGAN; DEMUTH, 2016): o *Levenberg-Marquardt backpropagation* e *Bayesian regularization backpropagation*. Na próxima sessão, será explicado o funcionamento desses algoritmos de aprendizagem assim como a implementação do GA.

3.1.1 Treinamento da Rede Neural usando um algoritmo de *backpropagation*

Vários problemas complexos podem ser resolvidos treinando MLPs de uma maneira supervisionada, com um algoritmo de aprendizado de *backpropagation* que é um processo de aprendizagem que acontece da seguinte maneira de acordo com (ROJAS, 1996). Inicialmente é aplicado na camada de entrada um conjunto de dados de entrada que se propagam, camada por camada através da rede para produzir um conjunto de sinais saídas. Durante esta etapa, todos os pesos sinápticos da rede são números reais fixados aleatoriamente. Em seguida, conforme aos sinais de erros resultantes dos valores estimados pela rede e os valores desejados, o algoritmo de aprendizagem minimiza uma função desses erros ajustando os pesos sinápticos a fim de fazer com que a resposta da rede se mova para mais perto da resposta desejada.

3.1.1.1 Levenberg-Marquardt backpropagation

O algoritmo de Levenberg-Marquardt se refere a uma variação do método de Newton que foi projetado para resolver o problema de quadrados mínimos não

linear (GAVIN, 2019). Conforme mencionado anteriormente, a aprendizagem ocorre por meio de ajuste dos pesos e para isso os algoritmos de aprendizagem minimizam um índice de performance. O algoritmo básico para a otimização numérica de um índice de performance é dado pela equação 7:

$$x_{k+1} = x_k - A_k^{-1} g_k \quad (7)$$

Onde $A_k \equiv \nabla^2 F(x) |_{x=x_k}$ é a matriz de peso e $g_k \equiv \nabla F(x) |_{x=x_k}$ é o negativo do gradiente.

A função de performance escolhido neste trabalho sendo o $F(x) = MSE$, pode-se reescrever $F(x)$ como sendo:

$$F(x) = \sum_{i=1}^N v_i^2(x) = v^T(x)v(x) \quad (8)$$

Então o elemento j^{th} do gradiente seria

$$[\nabla F(x)]_j = \frac{\partial F(x)}{\partial x_j} = 2 \sum_{i=1}^N v_i(x) \frac{\partial v_i(x)}{\partial x_j} \quad (9)$$

Portanto, o gradiente pode ser escrito em forma de matriz:

$$F(x) = \sum_{q=1}^Q (t_q - a_q)^T (t_q - a_q) = \sum_{i=1}^N (v_i)^2 \quad (10)$$

onde $e_{j,q}$ é o j ésimio elemento do erro para o par de entrada/destino q .

A etapa chave no algoritmo de Levenberg-Marquardt é o cálculo da matriz Jacobiana

$$\Delta F(x) = 2J^T(x)v(x) \quad (11)$$

onde a matriz Jacobiana é dada por

$$J = \begin{pmatrix} \frac{\delta v_1(x)}{\delta x_1} & \frac{\delta v_1(x)}{\delta x_2} & \dots & \frac{\delta v_1(x)}{\delta x_n} \\ \frac{\delta v_2(x)}{\delta x_1} & \frac{\delta v_2(x)}{\delta x_2} & \dots & \frac{\delta v_2(x)}{\delta x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\delta v_N(x)}{\delta x_1} & \frac{\delta v_N(x)}{\delta x_2} & \dots & \frac{\delta v_N(x)}{\delta x_n} \end{pmatrix} \quad (12)$$

O próximo passo consiste em descobrir a matriz Hessiana em que o elemento k, j da matriz Hessiana seria

$$[\nabla^2 F(x)]_{k,j} = \frac{\partial^2 F(x)}{\partial x_k \partial x_j} \quad (13)$$

$$[\nabla^2 F(x)]_{k,j} = 2 \sum_{i=1}^N \left\{ \frac{\partial v_i(x)}{\partial x_k} \frac{\partial v_i(x)}{\partial x_j} + v_i(x) \frac{\partial^2 v_i(x)}{\partial x_k \partial x_j} \right\} \quad (14)$$

Portanto pode-se aproximar a matrix Hessiana em forma matricial como sendo:

$$\Delta^2 F(x) = 2J^T(x)J(x) \quad (15)$$

O algoritmo de Levenberg-Marquardt dado pela equação 16 é então obtido substituindo as equações 15 e 11 na equação 7

$$\Delta x_k = -[J^T(x_k)J(x_k) + \mu I]^{-1} J^T(x_k)v(x_k) \quad (16)$$

A vantagem de se usar o algoritmo de Levenberg-Marquardt é que devido ao fator μ , esse algoritmo representa uma combinação do método do Gradiente Descendente e o método de Gauss-Newton. Durante o procedimento iterativo, os valores são atualizados usando o método Gauss-Newton se μ for pequeno e caso μ é grande, os valores são atualizados usando o método do do Gradiente Descendente (GAVIN, 2016). Essa característica confere ao algoritmo uma rapidez no treinamento de redes neurais artificiais, porém requer uma grande quantidade de memória (NELLES, 2013).

3.1.1.2 Bayesian regularization backpropagation

Primeiramente, a inferência bayesiana é um método de inferência estatística onde o teorema de Bayes é usado para atualizar a probabilidade de uma hipótese à medida que mais evidências ou informações se tornam disponíveis. Isso permite, como mostra a equação 17, derivar a probabilidade posterior como consequência de dois antecedentes: uma probabilidade anterior e uma função de probabilidade derivada de um modelo estatístico para os dados observados (HAGAN; DEMUTH; BEALE, 1997).

$$P(H | E) = \frac{P(E | H) \cdot P(H)}{P(E)} \quad (17)$$

onde H representa qualquer hipótese, $P(H)$ é a probabilidade anterior, E é a evidência, $P(H | E)$ é a probabilidade posterior, $P(E | H)$ é a probabilidade de observar E dado H e $P(E)$ às vezes é denominado a probabilidade marginal ou "evidência modelo".

A *Bayesian regularization backpropagation* visa a minimizar os erros de quadráticos e os pesos para determinar a combinação correta de modo a produzir uma rede que generalize bem. Ao aplicar o teorema de Bayes à RNA é função de probabilidade é definido como sendo:

$$P(x | D, \alpha, \beta, A) = \frac{P(D | x, \beta, A)P(x | \alpha, A)}{P(D | \alpha, \beta, A)} \quad (18)$$

Onde:

D representa o conjunto dos dados usados durante o treinamento,

x é um vetor que contém todos os pesos,

A representa a classe de arquitetura da rede e,

α, β são parâmetros associados às probabilidades de densidades $P(D | x, \beta, A) P(x | \alpha, A)$.

O índice de performance usado pelo algoritmo para treinar a RNA é dado pela soma quadrada dos pesos da rede E_W e a soma média dos erros de rede E_D .

$$F(x) = \beta E_D + \alpha E_W \quad (19)$$

Onde,

$$E_D = \sum_{i=1}^n e_i(n)^2 \quad (20)$$

e

$$E_W = \sum_{i=1}^n w_i^2 \quad (21)$$

Esse índice é então minimizado, usando o algoritmo de *Levenberg-Marquardt* e os parâmetros α, β estimados pela probabilidade de densidade $P(\alpha, \beta | D, A)$ até atingir a convergência. Esse procedimento faz com que a *Bayesian regularization backpropagation* seja mais tentada comparando com *Levenberg-Marquardt backpropagation*, porém mais robusto e menos susceptível para sofrer com *overfitting* (HAGAN; DEMUTH; BEALE, 1997).

3.1.2 Algoritmo Genético

Durante um processo de desenvolvimento de calibração, diferentes modelos caixa-preta do motor precisam ser desenvolvidos e como cada um desses modelos

são diferente um do outro por causa da suas complexidades ou número de variáveis, é necessário encontrar a arquitetura MIMO-RNA adequada para cada um deles. Por esse motivo, foi implementado o GA para otimizar a arquitetura MIMO-ANN.

O processo de otimização usando o GA, conforme mostrado no Algoritmo 1, começa gerando aleatoriamente o indivíduo inicial com os geradores de sequencias Quase-aleatório de Halton que é uma função dentro do ambiente MATLAB que pode ser chamado por *haltonset()*. Um indivíduo é representados por um conjunto de genes que são as informações que o definem como mostrado na figura 17. Para representação dos genes, foi usado uma codificação decimal onde os números de camadas, neurônios e épocas recebem os seus valores correspondentes e para cada uma das funções de ativação do conjunto $\{elliosig, logsig, radbas, tansig\}$ foram associados respectivamente aos números naturais no conjunto $\{1, 2, 3, 4\}$.

Figura 17 – Exemplo de genes de dois indivíduos e do descendente deles sem mutação de genes

	H. Laves Num	Nodes in each layer					Epoch	Activation function				
Mate 1	5	12	25	8	32	17	350	1	1	4	4	2
Mate 2	3	21	4	18	20	14	520	2	3	1	4	1
Offspring	3	21	25	8	32	14	520	2	3	4	4	1

Fonte: Autoria própria

Cada indivíduo é uma arquitetura do MIMO-RNA para modelagem do motor e possui a sua própria *fitness* que representa o seu nível de adaptação durante o processo. Uma vez que analisamos o desempenho das MIMO-RNAs pelo erro quadrático médio (MSE) e como $MSE \geq 0$, o *fitness* de cada indivíduo pode ser definida pela equação 22 onde os melhores indivíduos terão um *fitness* próximo de de 1 e os piores um *fitness* próximo do 0.

$$Fitness = \frac{1}{1 + MSE} \quad (22)$$

A próxima etapa é a seleção que consiste em selecionar os indivíduos que participarão do *crossover*. Para isso, implementou-se o método de seleção chamado seleção por classificação (do inglês *rank selection*). Nesse método de seleção, como ilustra a Figura 18 após classificar os indivíduos com base nos seus *fitness*, eles recebem uma classificação n , $n \in \{1, 2, \dots, N\}$, onde N é o tamanho da população. O indivíduo com melhor *fitness* recebe N , o pior rendimento 1, e então a probabilidade de um indivíduo i ser selecionado para participar do *crossover* é dada por:

$$p(i) = \frac{rank(i)}{n * (n - 1)} \quad (23)$$

A escolha desse método de seleção foi motivado pelo fato de ser robusto mantendo uma pressão constante na busca evolutiva (ANAND; AFREEN; YAZDANI, 2015).

Algoritmo 1 – Algoritmo genético

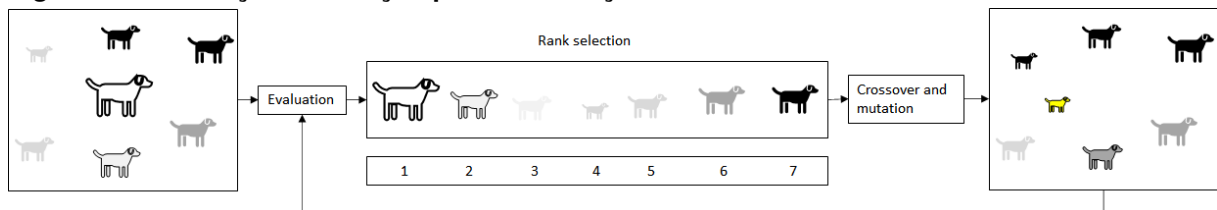
```

1: Inicialização
2: Definir o tamanho da população
3: Definir o percentual da população que fica para próxima geração
4: Definir a taxa de mutação
5: Gerar a população aleatoriamente usando gerador de sequencias Quase-aleatório de Halton
6: enquanto pop < tamanho da população faça
7:   Avaliar o fitness de cada indivíduo
8: finaliza enquanto
9: Melhor de todos ← melhor indivíduo
10: enquanto critérios de parada não são atendidos faça
11:   Avaliação
12:   Seleção por classificação
13:   Crossover
14:   pop ← descendentes
15:   Mutação
16:   melhor ← melhor individuo
17:   se melhor ≥ Melhor de todos então
18:     Melhor de todos ← melhor
19:   finaliza se
20: finaliza enquanto
21: Retorna ← Melhor de todos

```

Fonte: Autoria própria.

Figura 18 – Ilustração da seleção por classificação



Fonte: Autoria própria

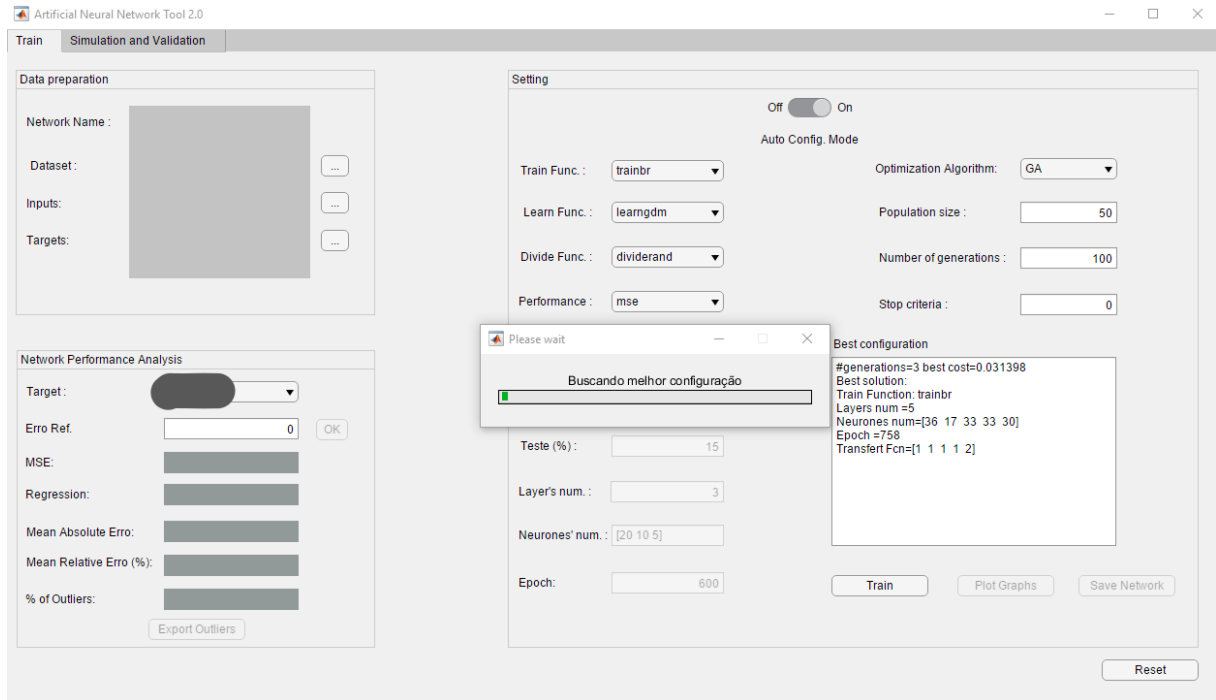
Uma vez que os pais são selecionados, os descendentes são gerados usando o operadores de *crossover* ponto a ponto e *mutação*. O *fitness* dos novos indivíduos é avaliada e os piores indivíduos da população são substituídos pelos melhores novos indivíduos.

3.2 INTERFACE GRÁFICA DA FERRAMENTA

A Figura 19 a seguir mostra a interface gráfica da ferramenta em modo de busca automática. A ferramenta foi desenvolvida com base nos princípios de programação orientada a objeto, dentro do ambiente MATLAB R2016b disponibilizado pela montadora e foi projetada com foco no cliente para que seja intuitivo, ágil e robusto. Toda metodologia para modelagem de motor a combustão interna explicada anteriormente foi implementada dentro da ferramenta para que ao criar um modelo, o cliente se preocupe apenas em carregar os dados, selecionar as variáveis do modelo e configurar os parâmetros da RNA ou lançar uma busca automática para encontrar uma

arquitetura ideal para o modelo a ser criado. Devido à questões de segredo industrial, os detalhes sobre o processo de desenvolvimento dessa ferramenta não serão apresentados neste trabalho.

Figura 19 – Ilustração da interface gráfica da ferramenta desenvolvida para criação de modelos de RNAs



Fonte: Autoria própria

4 RESULTADOS E DISCUSSÃO

Foi realizada em banco de ensaio de motores uma aquisição de dados de motor 1.0l de 3 cilindros com os seguintes pontos de operação: Rotação do motor variada de 1000 rpm a 5800 rpm, o VVT de admissão e escapamento variado do ângulos mínimo ao máximo. Em cada condição, uma rampa dinâmica lenta de pressão do coletor de admissão foi feita de 200hPa a 1000hPa para medir todo o campo do motor. O sistema de automação da bancada de teste controlou o ângulo da faísca de ignição para dar $CA50 = 8^\circ$ ou o valor mínimo que respeita os limites do motor de pressão do cilindro e detonação. O sistema de automação de bancada controlava também a injeção de combustível que fornece o $\lambda = 1$ ou o mínimo de enriquecimento de combustível possível para respeitar o limite de temperatura do motor de escape. Esta aquisição dinâmica permite reduzir consideravelmente o tempo de teste.

Esses pontos de aquisição de dados foram discretizados em um total de 16.627 pontos. A Tabela 3 mostra a visão geral dos pontos de aquisição, onde cada ponto "x" representa cerca de 43 pontos de aquisição da pressão de admissão de 200hPa a 1000hPa em intervalos de 20hPa. Alguns pontos não foram adquiridos devido à instabilidade de combustão do motor como baixa rotação do motor e posições extremas do VVT.

Tabela 3 – Visão geral dos pontos de aquisição

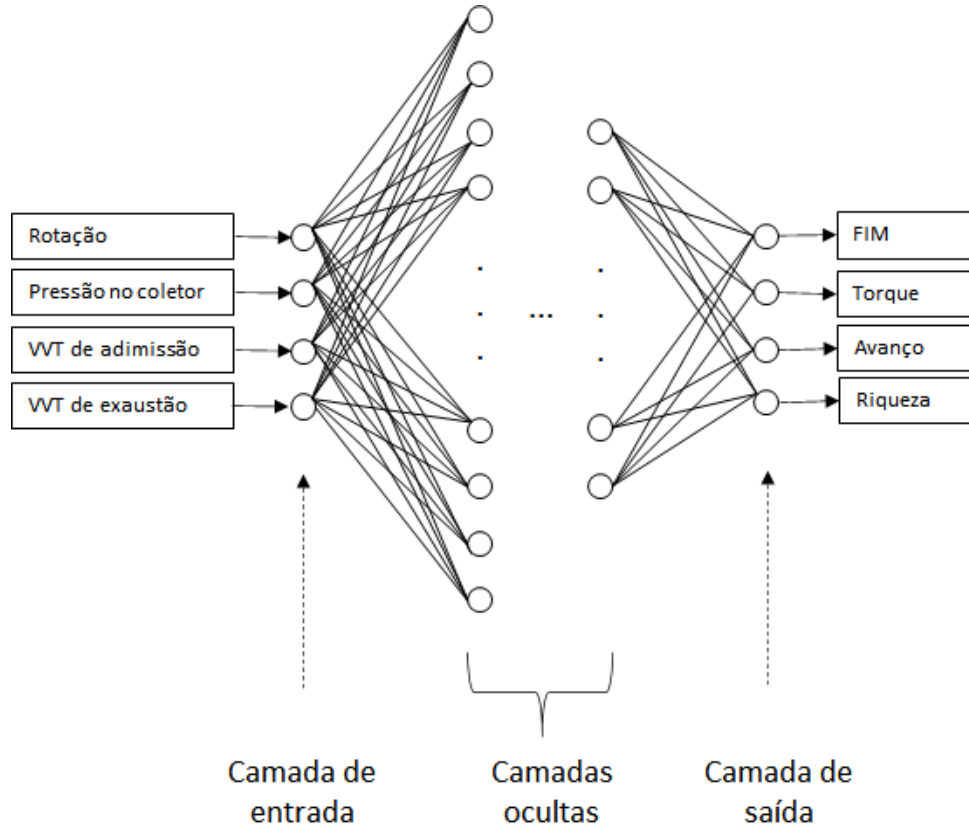
VVT de admissão	VVT de exaustão	Rotação														
		1000	1200	1400	1700	2200	2500	2700	3100	3600	4000	4300	4700	5100	5400	5800
0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	30	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
10	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	30	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
20	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	30	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
30	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	30	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
40	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	30	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
50	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	30	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Fonte: Autoria própria

Uma vez que os testes realizados na bancada de teste estão sempre em condições padrão de temperatura do ar de entrada (25°) e temperatura do líquido de

arrefecimento (90°C), os seguintes parâmetros foram usados como entrada para o modelo RNA do motor representado pela Figura 20: rotação, pressão de admissão, VVT de admissão e VVT de exaustão. As saídas do modelo são: torque, avanço, riqueza e a massa de injeção de combustível (FIM).

Figura 20 – Arquitetura da RNA para modelagem do motor



Fonte: Autoria própria

Para otimizar a arquitetura das RNAs, usou-se 43% dos dados adquiridos conforme mostra a Tabela 4, onde são representados em "X" os pontos de aquisição usados. Todos os dados de entrada da RNA foram padronizados entre 0 e 1 de acordo com a equação 2. O algoritmo genético foi configurado para 50 indivíduos por geração sendo que o percentual de indivíduos que permanecem para próxima geração é de 50%. A taxa de mutação foi configurado para 8%. Todas as simulações e resultados a seguir foram realizadas, usando a ferramenta desenvolvida durante esse trabalho, em um notebook com 32GB de ram e um processador intel i5-6446HD com clock em 3,08 GHz o tempo todo. Devido ao tempo de processamento, a otimização das arquiteturas usando o algoritmo de treinamento *Levenberg-Marquardt backpropagation* foi realizada sobre 16 gerações e 5 gerações para otimização das arquiteturas usando como algoritmo de treinamento o *Bayesian regularization backpropagation*.

A figura 21 mostra o gráfico boxplot da evolução do *fitness* das arquiteturas treinadas usando o algoritmo *Levenberg-Marquardt backpropagation*. Observa-se que

Tabela 4 – Redução de 57% dos dados do motor

VVT admissão	VVT Exaustão	Rotação															
		1000	1200	1400	1700	2200	2500	2700	3100	3600	4000	4300	4700	5100	5400	5800	
0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
	10		X		X		X		X		X		X		X		
	20	X		X		X		X		X		X		X		X	
	30		X		X		X		X		X		X		X		
	40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
10	0		X		X		X		X		X		X		X		
	10				X				X				X				
	20		X				X				X				X		
	30				X				X				X				
	40		X		X		X		X		X		X		X		
20	0	X		X		X		X		X		X		X		X	
	10	X				X				X				X			
	20			X				X				X				X	
	30	X				X				X				X			
	40	X		X		X		X		X		X		X		X	
30	0		X		X		X		X		X		X		X		
	10				X				X				X				
	20		X				X				X				X		
	30				X				X				X				
	40		X		X		X		X		X		X		X		
40	0	X		X		X		X		X		X		X		X	
	10	X				X				X				X			
	20			X				X				X				X	
	30	X				X				X				X			
	40	X		X		X		X		X		X		X		X	
50	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
	10		X		X		X		X		X		X		X		
	20	X		X		X		X		X		X		X		X	
	30		X		X		X		X		X		X		X		
	40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

Fonte: Autoria própria

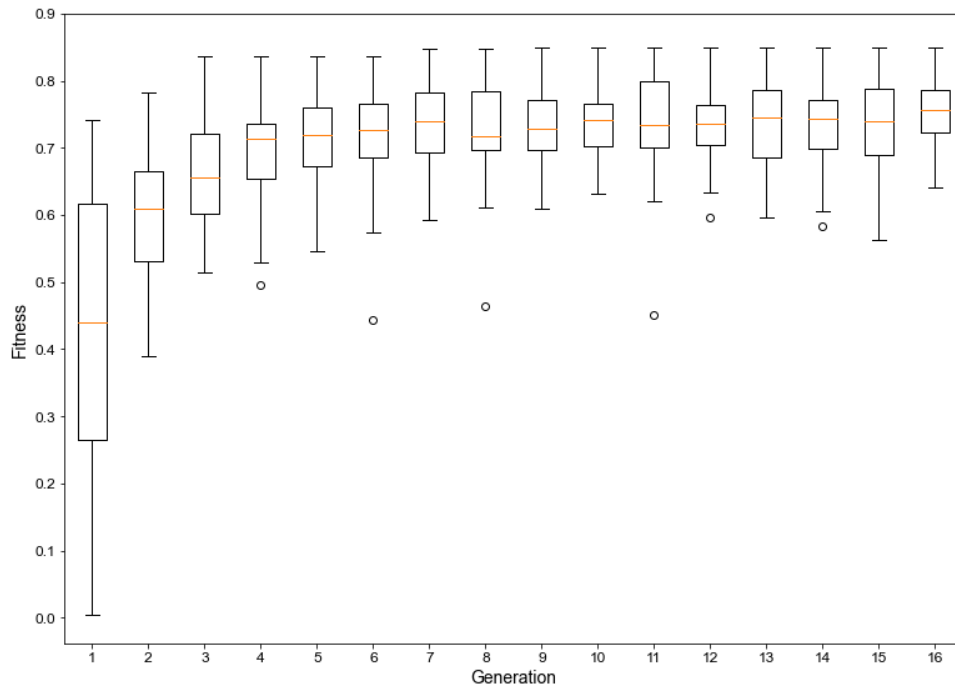
houve uma estabilização na dispersão do *fitness* dos indivíduos a partir da quinta geração sendo que o maior *fitness* dentro dessas gerações foi obtido na nona geração com um *fitness* igual a 0,849098258. A maior dispersão de *fitness* foi observada na geração inicial onde a pior configuração encontrada teve um *fitness* igual a 0,004287431 enquanto a melhor configuração dentro dessa geração teve um *fitness* igual a 0,742159365.

As figuras 24(a) e 24(b) apresentam respectivamente o histograma de erros e a regressão da melhor configuração de MIMO-RNA durante a fase de treinamento usando o *Levenberg-Marquardt backpropagation*. Esses gráficos mostram que a rede neural correlacionou muito bem as variáveis de entrada da rede com as de saída, pois a regressão geral foi de 0,99842. Observa-se também pelo histograma dos erros que a grande maioria dos erros ficaram próximo do 0 demonstrando que o algoritmo foi eficaz no quesito de minimizar o índice de performance com base no é realizado o ajuste dos pesos da RNA, isto é, a aprendizagem.

A figura 23 mostra a evolução do *fitness* das arquiteturas treinadas usando o algoritmo *Bayesian regularization backpropagation*. Observa-se uma tendência de convergência a partir da quinta geração, onde foi encontrado também o melhor indivíduo com um *fitness* igual a 0,976427097. A maior dispersão de *fitness* usando esse algoritmo foi observada também na geração inicial onde o pior indivíduo teve um *fitness* igual a $4,65373 * 10^{-07}$ enquanto o melhor indivíduo teve um *fitness* igual a 0,962113296.

As figuras 24(a) e 24(a) apresentam respectivamente o histograma de erros e a regressão da melhor arquitetura de MIMO-RNA durante a fase de treinamento

Figura 21 – Gráfico Box plot da evolução do *fitness* da população usando o algoritmo *Levenberg-Marquardt backpropagation*



Fonte: Autoria própria

Tabela 5 – Melhor configuração encontrada para cada algoritmo de treinamento

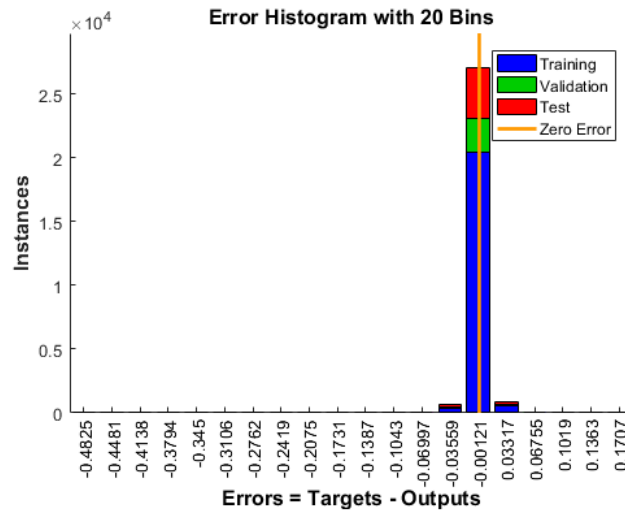
Algoritmo	Melhor configuração				Performance	
	Número de camadas	Número de neurônios	Número de épocas	Função de ativação em cada camada	MES	Fitness
Levenberg-Marquardt	5	33 - 32 - 39 - 17 - 33	49	tansig nas quatro primeira camada e radial basis na quinta camada	0,1777	0,849113
Bayesian regularization	5	33 - 32 - 39 - 38 - 30	758	Elliot symmetric em todas as camadas	0,0241	0,976467

Fonte: Autoria própria

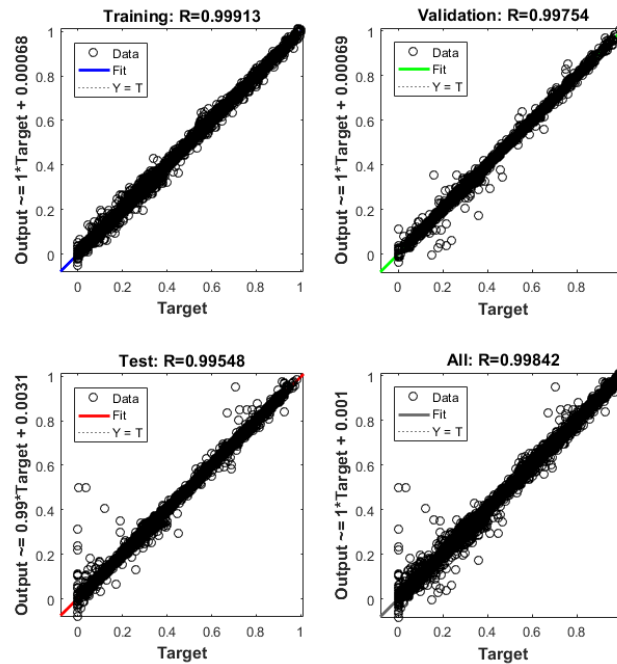
usando o *Bayesian regularization backpropagation*. A regressão geral foi 0,99901, o que mostra que a rede neural correlacionou muito bem as variáveis de entrada da rede com as de saída. Observa-se também pelo histograma dos erros que a grande maioria dos erros ficaram próximo do 0 demonstrando que o algoritmo foi eficaz no quesito de minimizar o índice de performance com base no é realizado o ajuste dos pesos da RNA, isto é, a aprendizagem.

Na tabela 5 está apresentado a melhor configuração de RNA encontrada para cada algoritmo de treinamento, o *fitness* e o MSE das RNAs nessas configurações. O tempo que o GA levou para para completar as 16 gerações fazendo a busca pela melhor configuração usando o *Levenberg-Marquardt backpropagation* foi de 25 horas e 12 minutos enquanto o tempo levado para completar as 5 gerações usando o *Bayesian regularization backpropagation* foi de 203 horas e 43 minutos. Esses resultados mostra conforme esperado, que o algoritmo *Levenberg-Marquardt backpropagation* é mais rápido do que *Bayesian regularization backpropagation*, porém, teve uma taxa de precisão menor apesar ter sido simulado sobre um número de gerações maior do que para o *Bayesian regularization backpropagation*.

Figura 22 – Histograma de erros e a regressão durante o treinamento da RNA usando o algoritmo *Levenberg-Marquardt backpropagation*



(a)



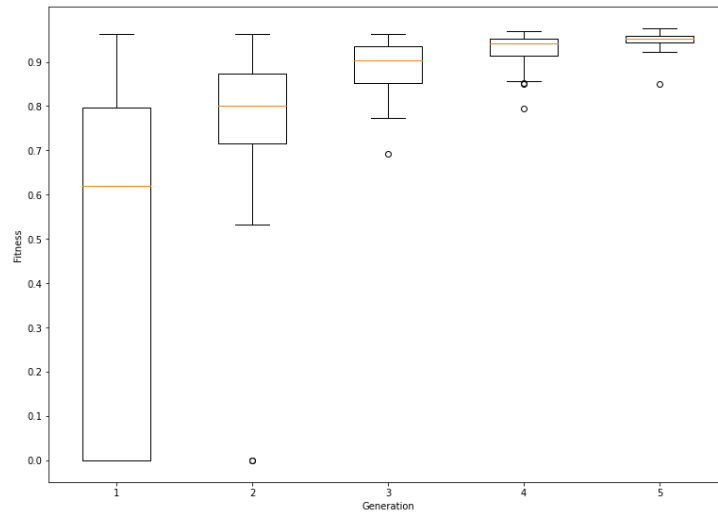
(b)

Fonte: Autoria própria

Para avaliar a capacidade de generalização das melhores configurações encontradas, foi realizada uma simulação com todos os pontos de aquisição do motor usando, isto é, incluindo os 57% dos pontos que não foram usados durante o processo de treinamento das RNAs. O objetivo é de calcular o percentual de pontos com erro dentro do critério de aceitação de erros de estimação estabelecida pelos engenheiros de calibração da montadora conforme mostra a tabela 6.

A tabela 7 mostra o resultado dessa simulação usando a RNA criada a partir da melhor configuração encontrada para o algoritmo *Levenberg-Marquardt backpro-*

Figura 23 – Gráfico Box plot da evolução do fitness da população usando o algoritmo *Bayesian regularization backpropagation*



Fonte: Autoria própria

Tabela 6 – Critérios de validação das Redes Neurais

Variável	Máximo erro absoluto desejado	Máximo erro relativo desejado
FIM	NA	<5%
Torque	<3Nm	NA
Avanço	<1,25°	NA
Riqueza	<0,015	NA

Fonte: Autoria Própria

pagation. O percentual de pontos que ficaram dentro dos critérios de aceitação para o (FIM), avanço de ignição, riqueza e torque foram de 76,73%, 64,80%, 95,79% e 80,65% respectivamente.

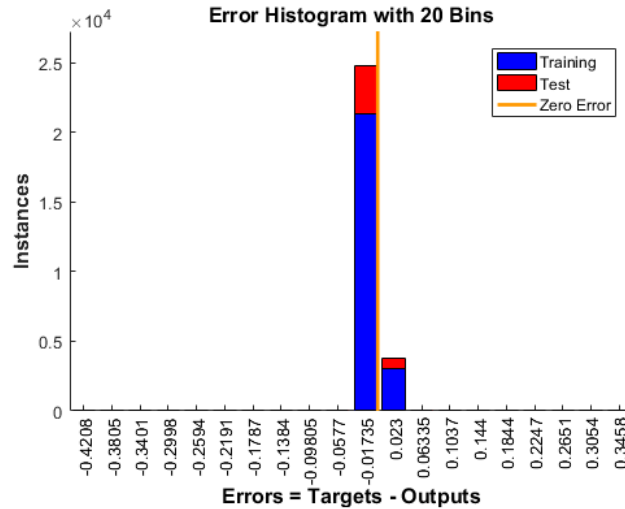
Tabela 7 – Precisão da RNA treinada com 43% dos pontos usando o Levenberg-Marquardt backpropagation

Variáveis	Estimação de erro					
	Erro relativo	<1%	<2%	<3%	<4%	<5%
FIM	% de pontos	36,93992	53,77398	63,88404	71,05912	76,73062
	Erro absoluto	<0.25°	<0.5°	<0.75°	<1°	<1.25°
Avanço de Ignição	% de pontos	26,17429	43,23089	53,50334	60,13713	64,80423
	Erro absoluto	<0,005	<0,0075	<0,01	<0,0125	<0,015
Riqueza	% de pontos	82,29386	89,79972	92,84898	94,76755	95,79599
	Erro absoluto	<1Nm	<1,5Nm	<2Nm	<2,5Nm	<3Nm
Torque	% de pontos	55,30162	65,42371	72,08155	77,00126	80,65797

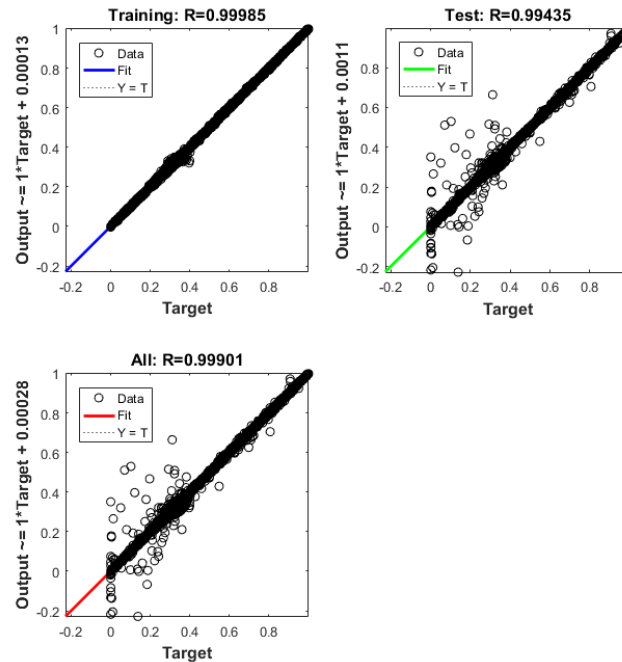
Fonte: Autoria própria

A tabela 8 mostra o resultado dessa simulação usando a RNA criada a partir da melhor configuração encontrada para o algoritmo *Bayesian regularization backpro-*

Figura 24 – Histograma de erros e a regressão durante o treinamento da RNA usando o algoritmo *Bayesian regularization backpropagation*



(a)



(b)

Fonte: Autoria própria

pagation. O percentual de pontos que ficaram dentro dos critérios de aceitação para o (FIM), avanço de ignição, riqueza e torque foram de 94,46%, 84,75%, 94,05% e 96,49% respectivamente.

De acordo com esse resultados, observa-se que o algoritmo *Bayesian regularization backpropagation* possui uma capacidade aprendizagem e por tanto de generalização superior ao o algoritmo *Levenberg-Marquardt backpropagation*.

Em seguida, foi realizada outra simulação mas dessa vez com RNAs treinadas com 65% dos pontos de aquisição do motor, conforme mostra a Tabela 11, com

Tabela 8 – Precisão da RNA treinada com 43% dos pontos usando a Bayesian regularization backpropagation

Variáveis	Estimação de erro					
	Erro relativo	<1%	<2%	<3%	<4%	<5%
FIM	% de pontos	67,81139	83,76737	89,75161	92,59638	94,46683
	Erro absoluto	<0.25°	<0.5°	<0.75°	<1°	<1.25°
Avanço de Ignição	% de pontos	49,87069	68,72557	76,39983	81,32555	84,75371
	Erro absoluto	<0,005	<0,0075	<0,01	<0,0125	<0,015
Riqueza	% de pontos	82,87123	88,96373	91,57996	92,96927	94,05184
	Erro absoluto	<1Nm	<1,5Nm	<2Nm	<2,5Nm	<3Nm
Torque	% de pontos	84,34474	90,57557	93,6128	95,47122	96,49365

Fonte: Autoria própria

as melhores configuração de redes encontradas. Os resultados obtidos são apresentados nas tabelas 9 e 10. O percentual de pontos que ficaram dentro dos critérios de aceitação para o (FIM), avanço de ignição, riqueza e torque, pela RNA treinado com o algoritmo *Levenberg-Marquardt backpropagation*, foram de 90,11%, 81,30%, 97,61% e 93,07% respectivamente. Já o percentual de pontos que ficaram dentro dos critérios de aceitação para o (FIM), avanço de ignição, riqueza e torque, pela RNA treinado com o algoritmo *Bayesian regularization backpropagation*, foram de 99,31%, 96,16%, 97,93% e 99,41% respectivamente.

Tabela 9 – Precisão da RNA treinada com 65% dos pontos usando o Levenberg-Marquardt backpropagation

Variáveis	Estimação de erro					
	Erro relativo	<1%	<2%	<3%	<4%	<5%
FIM	% de pontos	53,40711	72,99573	81,13911	86,41366	90,10645
	Erro absoluto	<0.25°	<0.5°	<0.75°	<1°	<1.25°
Avanço de Ignição	% de pontos	32,59758	55,65045	68,89397	76,67048	81,29548
	Erro absoluto	<0,005	<0,0075	<0,01	<0,0125	<0,015
Riqueza	% de pontos	89,22836	94,13003	95,9644	97,05299	97,61833
	Erro absoluto	<1Nm	<1,5Nm	<2Nm	<2,5Nm	<3Nm
Torque	% de pontos	72,45444	82,54646	87,95333	91,11084	93,0655

Fonte: Autoria própria

Os gráficos das figuras 25, 26, 27, 28, 29 e 30 mostram um comparativo entre os valores real e valor simulado dentro de algumas regiões do motor assim como os sinais de erro usando a melhor configuração de RNA criada a partir de 65% dos dados usando o algoritmo *Bayesian regularization backpropagation*.

Tabela 10 – Precisão da RNA treinada com 65% dos pontos usando a Bayesian regularization backpropagation

Variáveis	Estimação de erro					
	Erro relativo	<1%	<2%	<3%	<4%	<5%
FIM	% de pontos	85,31906	95,05022	97,79275	98,86329	99,30835
	Erro absoluto	<0.25°	<0.5°	<0.75°	<1°	<1.25°
Avanço de Ignição	% de pontos	60,5882	83,41252	91,18903	94,58712	96,16888
	Erro absoluto	<0,005	<0,0075	<0,01	<0,0125	<0,015
Riqueza	% de pontos	92,99332	95,68774	96,86654	97,4379	97,93709
	Erro absoluto	<1Nm	<1,5Nm	<2Nm	<2,5Nm	<3Nm
Torque	% de pontos	95,74788	98,11151	98,92945	99,23618	99,4106

Fonte: Autoria própria

Tabela 11 – Redução em 35% dos dados do motor

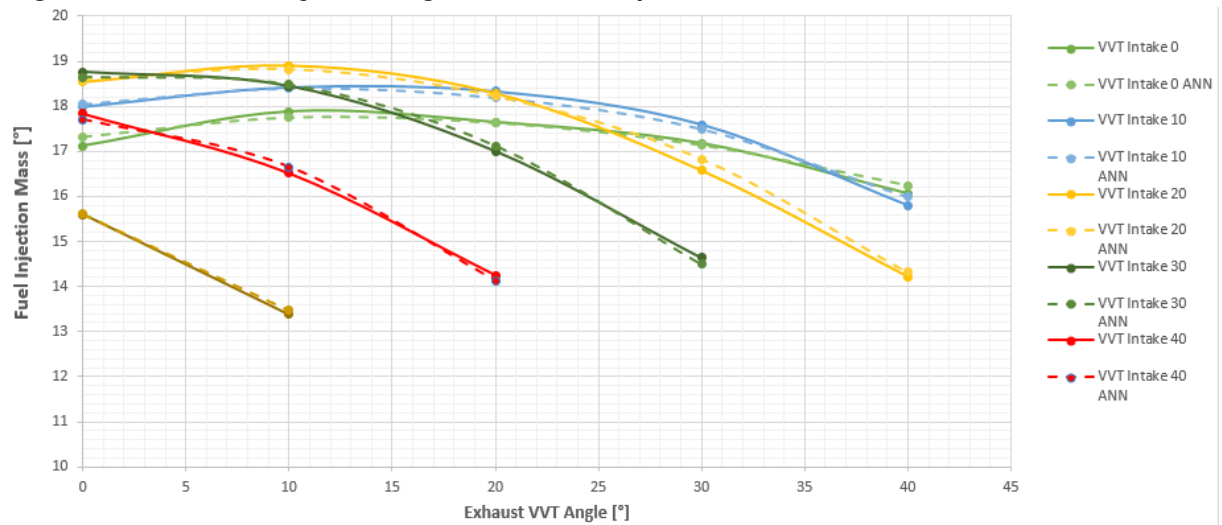
VVT de admissão	VVT Exaustão	Rotação														
		1000	1200	1400	1700	2200	2500	2700	3100	3600	4000	4300	4700	5100	5400	5800
0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	30	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
10	0		X		X		X		X		X		X		X	
	10		X		X		X		X		X		X		X	
	20		X		X		X		X		X		X		X	
	30		X		X		X		X		X		X		X	
	40		X		X		X		X		X		X		X	
20	0	X		X		X		X		X		X		X		X
	10	X		X		X		X		X		X		X		X
	20	X		X		X		X		X		X		X		X
	30	X		X		X		X		X		X		X		X
	40	X		X		X		X		X		X		X		X
30	0		X		X		X		X		X		X		X	
	10		X		X		X		X		X		X		X	
	20		X		X		X		X		X		X		X	
	30		X		X		X		X		X		X		X	
	40		X		X		X		X		X		X		X	
40	0	X		X		X		X		X		X		X		X
	10	X		X		X		X		X		X		X		X
	20	X		X		X		X		X		X		X		X
	30	X		X		X		X		X		X		X		X
	40	X		X		X		X		X		X		X		X
50	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	30	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Fonte: Autoria própria

A figura 25 mostra o gráfico da massa de combustível injetada (FIM), em função dos ângulos VVT em 1700 rpm e 540 mbar, onde as linhas cheias representam os valores estimados pelo modelo do motor baseado no RNA e as linhas cheias os valores verdadeiros medidos. Obversa-se o modelo virtual do motor acompanha perfeitamente o comportamento real do motor.

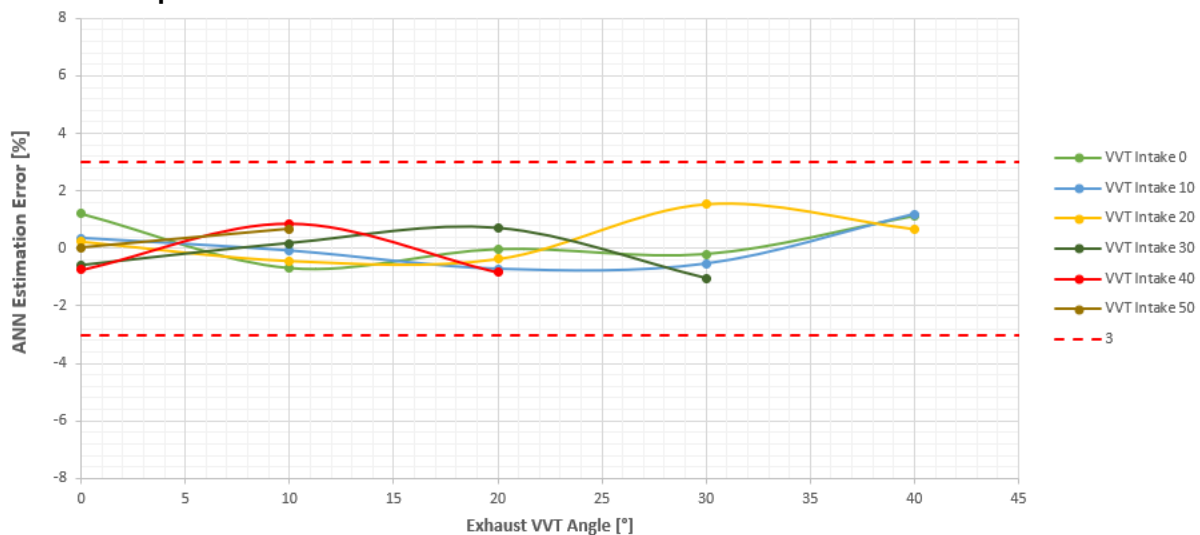
O gráfico da Figura 26 representa o sinal de erro na estimação feita pela RNA do FIM em função do ângulo VVT em 1700 rpm e 540 mbar. Percebe-se que o erro ficou dentro dos limites de erros aceitável durante a calibração que foi definido para um erro absoluto máximo de 3% .

Figura 25 – FIM em função do ângulo VVT - 1700 rpm/540 mbar



Fonte: Autoria própria

Figura 26 – Sinal de erro na estimação feita pela RNA do FIM em função do ângulo VVT - 1700 rpm/540 mbar

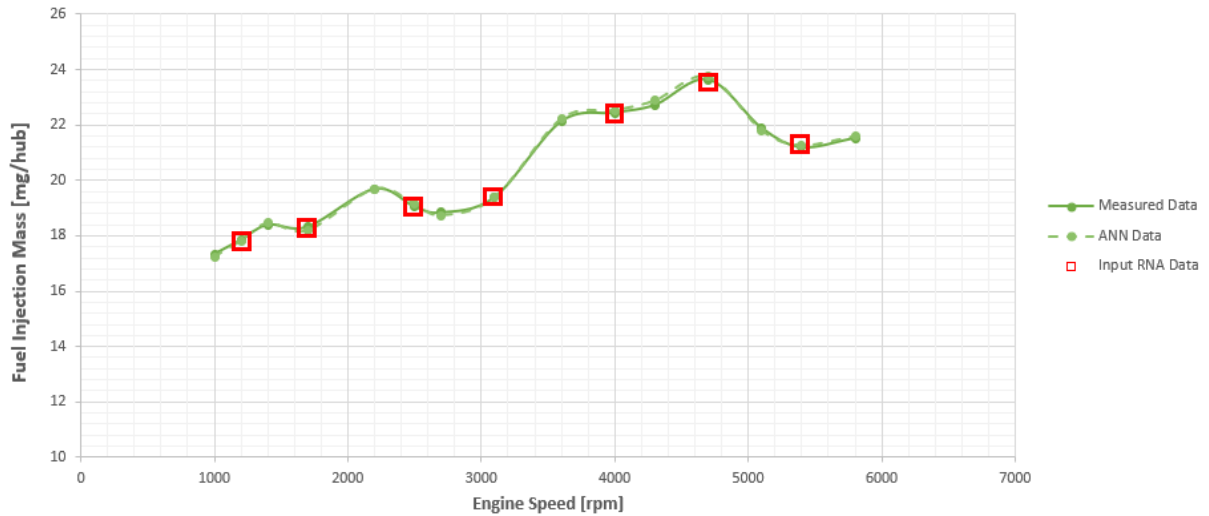


Fonte: Autoria própria

O gráfico da Figura 27 mostra o FIM em função da rotação em VVTIN 10, VVTEX 20 e 540 mbar, onde são representados em vermelho os pontos que foram usado nessa região durante o treinamento. Pode-se observar que o modelo da RNA foi fiel ao modelo físico do motor mesmo nos pontos que não foram usados durante o treinamento da RNA. Além disso, nota-se a grande capacidade da RNA de aprender, pois, apesar da complexidade dessa curva a RNA conseguiu fazer estimações dentro dos critérios de erro máximo como mostra a Figura 28

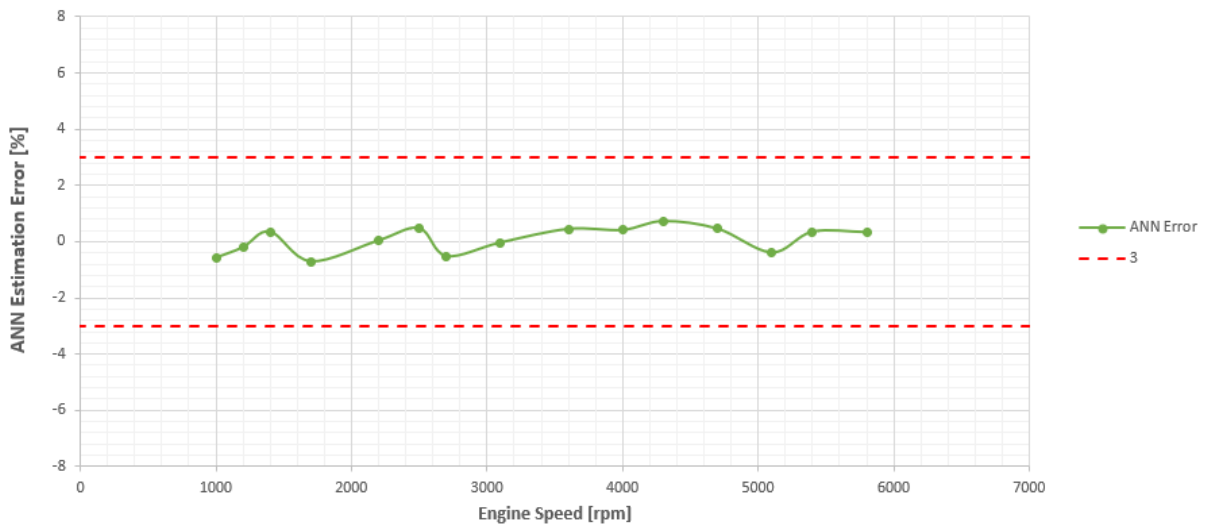
A Figura 29 mostra o gráfico do FIM em função da pressão de admissão em 1700rpm, VVTIN 20 e VVTEX 20. Nessa região, nenhum dado foi usado para criação do modelo virtual do motor e pode-se observar que o comportamento do modelo virtual

Figura 27 – FIM em função da rotação em VVTIN 10 / VVTEX 20 / 540mbar



Fonte: Autoria própria

Figura 28 – Sinal de erro na estimação feita pela RNA do FIM em função da rotação em VVTIN 10 / VVTEX 20 / 540mbar

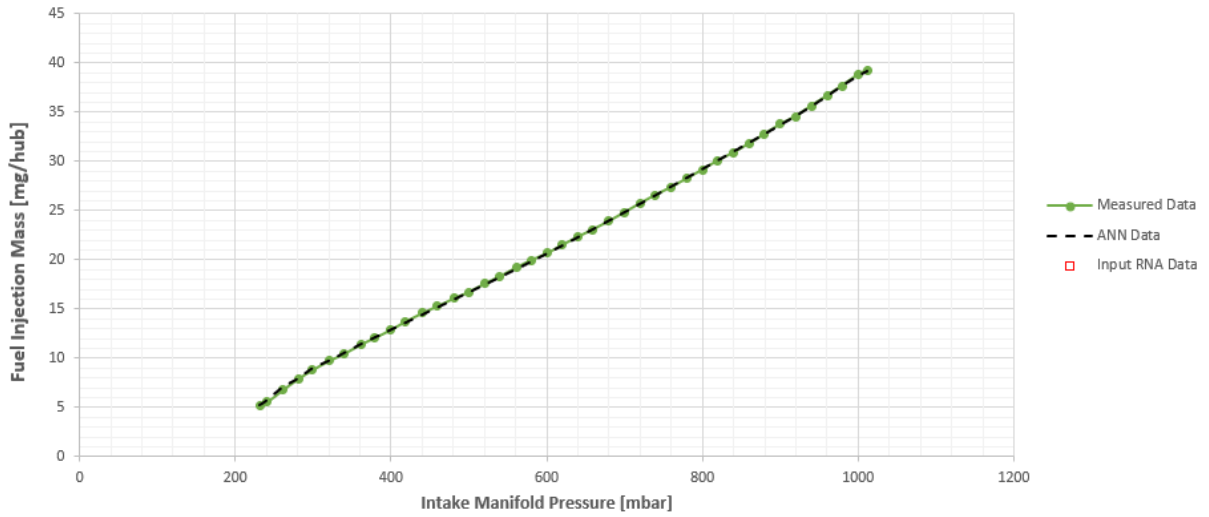


Fonte: Autoria própria

do motor acompanha perfeitamente o comportamento real do motor. Porém, de acordo como o sinal de erro mostrada na Figura 30, os erros do modelo virtual passaram dos limites de aceitação em pressões baixas.

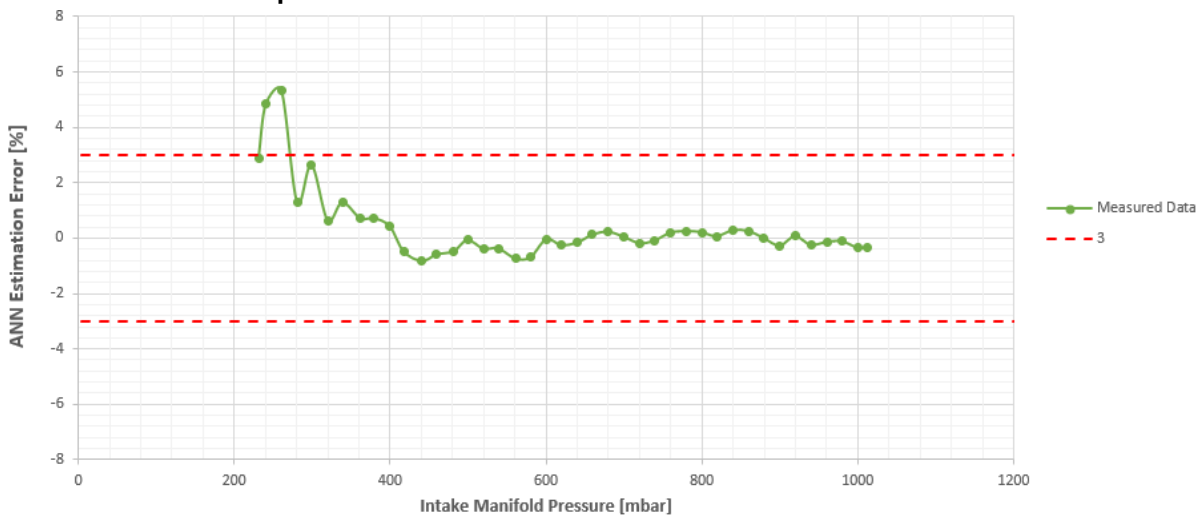
Com base nos fatos expostos nesse capítulo, percebe-se o potencial da ferramenta desenvolvida durante o presente trabalho para criação de modelos virtual de motores de combustão interna usando técnicas de redes neurais artificiais.

Figura 29 – FIM em função da pressão de admissão em 1700rpm / VVTIN 20 / VVTEX 20



Fonte: Autoria própria

Figura 30 – Sinal de erro na estimação feita pela RNA FIM em função da pressão de admissão em 1700rpm / VVTIN 20 / VVTEX 20



Fonte: Autoria própria

5 CONCLUSÕES E PERSPECTIVAS

O presente trabalho tinha como objetivo de desenvolver uma ferramenta de simulação numérica executável para modelar virtualmente alguns parâmetros de motores de combustão interna do ciclo Otto usando Redes Neurais Artificiais. Para isso, foram investigados dois algoritmos de treinamento de redes neurais, (*Levenberg-Marquardt backpropagation and Bayesian regularization backpropagation*) disponibilizado pelo Matlab Neural Network Toolbox. Para encontrar uma arquitetura de RNA que proporciona um modelo virtual do motor com uma taxa de estimação aceitável das variáveis a serem modelados, foi implementado o algoritmo genético.

Para avaliar o potencial da ferramenta desenvolvida, os seguintes parâmetros foram usados como entrada para o modelo RNA: rotação, pressão de admissão, VVT de admissão e VVT de exaustão. As saídas do modelo são: torque, avanço, riqueza e a massa de injeção de combustível (FIM). Os resultados obtidos mostram que a ferramenta desenvolvida tem um grande potencial para desenvolvimento de modelos virtuais de motores de combustão interna, pois, usando apenas 43% dos dados de todo campo do motor para treinar a RNA, o melhor modelo encontrado ao simular todo campo do motor atendeu em 94,46%, 84,75%, 94,05% e 96,49% os critérios de aceitação para o (FIM), avanço de ignição, riqueza e torque respectivamente.

Como trabalhos futuros, pode-se em primeiro lugar implementar outros métodos de seleção dentro do GA para avaliar o método que apresenta o melhor desempenho. Pode ser também implementados outros tipos de algoritmos meta-heurísticos como o PSO e o DE para otimização da arquitetura das redes neurais para determinar o algoritmo que apresenta melhor resultado e comparar também os seus respectivos custos computacionais.

REFERÊNCIAS

AGGARWAL, Charu C et al. **Neural networks and deep learning**. [S.l.]: Springer, 2018. Citado 2 vezes nas páginas 25 e 26.

ANAND, Smit; AFREEN, Nishat; YAZDANI, Shama. A novel and efficient selection method in genetic algorithm. **International Journal of Computer Applications**, Citeseer, v. 129, n. 15, p. 7–12, 2015. Citado na página 41.

ANFAVEA. **Anuário da Indústria Automobilística Brasileira 2020**. 2020. Disponível em: <www.anfavea.com.br/anuario2020/anuario.pdf>. Citado na página 14.

ASHOK, B; ASHOK, S Denis; KUMAR, C Ramesh. A review on control system architecture of a si engine management system. **Annual Reviews in Control**, Elsevier, v. 41, p. 94–118, 2016. Citado 2 vezes nas páginas 22 e 23.

ATKINSON, Chris; MOTT, Gregory. **Dynamic model-based calibration optimization: An introduction and application to diesel engines**. [S.l.], 2005. Citado na página 23.

BEALE, MH; HAGAN, MT; DEMUTH, HB. Matlab r2016b: Neural network toolbox: User's guide. **The MathWorks Inc., Natick, MA**, 2016. Citado na página 37.

BENARDOS, PG; VOSNIAKOS, G-C. Optimizing feedforward artificial neural network architecture. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 20, n. 3, p. 365–382, 2007. Citado na página 24.

BRUNETTI, Franco. **Motores de Combustão Interna-Vol. 1**. [S.l.]: Editora Blucher, 2018. Citado na página 18.

CARVALHO, Adenilson R; RAMOS, Fernando M; CHAVES, Antonio A. Metaheuristics for the feedforward artificial neural network (ann) architecture optimization problem. **Neural Computing and Applications**, Springer, v. 20, n. 8, p. 1273–1284, 2011. Citado 2 vezes nas páginas 24 e 33.

CARVALHO, Rogério N de et al. A nova gasolina s50 e o proconve l6. **Blucher Engineering Proceedings**, v. 1, n. 1, p. 184–193, 2013. Citado na página 14.

CHENG, Min-Yuan; PRAYOGO, Doddy. Symbiotic organisms search: a new metaheuristic optimization algorithm. **Computers & Structures**, Elsevier, v. 139, p. 98–112, 2014. Citado 2 vezes nas páginas 32 e 33.

CONAMA. **RESOLUÇÃO Nº 492, DE 20 DE DEZEMBRO DE 2018**. 2018. Disponível em: <https://www.in.gov.br/materia/-/asset_publisher/Kujrw0TZC2Mb/content/id/56643907/do1-2018-12-24-resolucao-n-492-de-20-de-dezembro-de-2018-56643731>. Citado na página 14.

DORIGO, Marco; BIRATTARI, Mauro; STUTZLE, Thomas. Ant colony optimization. **IEEE computational intelligence magazine**, IEEE, v. 1, n. 4, p. 28–39, 2006. Citado 2 vezes nas páginas 32 e 33.

EZUGWU, Absalom E et al. A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems. **Neural Computing and Applications**, Springer, v. 32, n. 10, p. 6207–6251, 2020. Citado na página 32.

FISTER, Iztok et al. A comprehensive review of firefly algorithms. **Swarm and Evolutionary Computation**, Elsevier, v. 13, p. 34–46, 2013. Citado 2 vezes nas páginas 32 e 33.

FLÓREZ, Jesús Andrés Álvarez; AGRAMUNT, Ismael Callejón. **Máquinas térmicas motoras. Vol. 1**. [S.l.]: Edicions UPC, 2002. Citado 2 vezes nas páginas 18 e 19.

GAVIN, Henri. The levenberg-marquardt method for nonlinear least squares curve-fitting problems. 2011. **Available Online from: <http://people.duke.edu/~hpgavin/ce281/lm.pdf>**..(Accessed 15 Sept 2015), 2016. Citado na página 39.

GAVIN, Henri P. The levenberg-marquardt algorithm for nonlinear least squares curve-fitting problems. **Department of Civil and Environmental Engineering, Duke University <http://people.duke.edu/~hpgavin/ce281/lm.pdf>**, p. 1–19, 2019. Citado na página 38.

GONZALEZ, Ernesto Gutierrez; FLOREZ, Jesus Alvarez; ARAB, Sebastián. Development of the management strategies of the ecu for an internal combustion engine: computer simulation. **Mechanical Systems and Signal Processing**, Elsevier, v. 22, n. 6, p. 1356–1373, 2008. Citado na página 21.

HAGAN, Martin T; DEMUTH, Howard B; BEALE, Mark. **Neural network design**. [S.l.]: PWS Publishing Co., 1997. Citado 2 vezes nas páginas 39 e 40.

HAO, Duan et al. Experimental study of hydrogen enriched compressed natural gas (hcng) engine and application of support vector machine (svm) on prediction of engine performance at specific condition. **International Journal of Hydrogen Energy**, Elsevier, v. 45, n. 8, p. 5309–5325, 2020. Citado na página 32.

HAYKIN, Simon. **Neural networks: a comprehensive foundation**. [S.l.]: Prentice-Hall, Inc., 2007. Citado 3 vezes nas páginas 25, 27 e 28.

HE, Y; RUTLAND, CJ. Application of artificial neural networks in engine modelling. **International Journal of Engine Research**, SAGE Publications Sage UK: London, England, v. 5, n. 4, p. 281–296, 2004. Citado na página 32.

HENDRICKS, Elbert. The analysis of mean value engine models. **SAE Transactions**, JSTOR, p. 972–985, 1989. Citado na página 29.

HEYWOOD, John B. Combustion engine fundamentals. **1ª Edição. Estados Unidos**, 1988. Citado 2 vezes nas páginas 20 e 21.

HOLLAND, John H. Genetic algorithms. **Scientific american**, JSTOR, v. 267, n. 1, p. 66–73, 1992. Citado 2 vezes nas páginas 32 e 33.

ISERMANN, Rolf. Engine modeling and control. **Berlin: Springer Berlin Heidelberg**, Springer, v. 1017, 2014. Citado 4 vezes nas páginas 21, 22, 28 e 29.

ISERMANN, Rolf; HAFNER, Michael. Mechatronic combustion engines—from modeling to optimal control. **European Journal of control**, Elsevier, v. 7, n. 2-3, p. 220–247, 2001. Citado na página 23.

ISERMANN, Rolf; SEQUENZ, Heiko. Model-based development of combustion-engine control and optimal calibration for driving cycles: general procedure and application. **IFAC-PapersOnLine**, Elsevier, v. 49, n. 11, p. 633–640, 2016. Citado na página 24.

ISLAM, Badar ul et al. Optimization of neural network architecture using genetic algorithm for load forecasting. In: IEEE. **2014 5th International Conference on Intelligent and Advanced Systems (ICIAS)**. [S.l.], 2014. p. 1–6. Citado na página 24.

JUNG, Daniel. **Residual Generation Using Physically-Based Grey-Box Recurrent Neural Networks For Engine Fault Diagnosis**. 2020. Citado na página 29.

KARABOGA, Dervis; BASTURK, Bahriye. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. **Journal of global optimization**, Springer, v. 39, n. 3, p. 459–471, 2007. Citado 2 vezes nas páginas 32 e 33.

KASABOV, Nikola K. **Foundations of neural networks, fuzzy systems, and knowledge engineering**. [S.l.]: Marcel Alencar, 1996. Citado 2 vezes nas páginas 25 e 27.

KIENCKE, Uwe; NIELSEN, Lars. **Automotive control systems: for engine, driveline, and vehicle**. [S.l.]: IOP Publishing, 2000. Citado na página 23.

MAARANEN, Heikki; MIETTINEN, Kaisa; MÄKELÄ, Marko M. Quasi-random initial population for genetic algorithms. **Computers & Mathematics with Applications**, Elsevier, v. 47, n. 12, p. 1885–1895, 2004. Citado na página 36.

MEHRABIAN, Ali Reza; LUCAS, Caro. A novel numerical optimization algorithm inspired from weed colonization. **Ecological informatics**, Elsevier, v. 1, n. 4, p. 355–366, 2006. Citado 2 vezes nas páginas 32 e 33.

MILHOR, Carlos Eduardo. Sistema de desenvolvimento para controle eletrônico dos motores de combustão interna ciclo otto. **São Carlos. 72p. Dissertação de Mestrado-Escola de Engenharia de São Carlos, Universidade de São Paulo**, 2002. Citado 2 vezes nas páginas 18 e 19.

MOHAMED, Zahraa E. Using the artificial neural networks for prediction and validating solar radiation. **Journal of the Egyptian Mathematical Society**, Springer, v. 27, n. 1, p. 47, 2019. Citado 2 vezes nas páginas 23 e 24.

MORANM, J; HN, BOETTNERDD SHAPIRO et al. **Fundamentals of engineering thermodynamics**. [S.l.]: Hoboken: JohnWiley&Sons, 2010. Citado 2 vezes nas páginas 19 e 20.

NAJAFI, G et al. Svm and anfis for prediction of performance and exhaust emissions of a si engine with gasoline–ethanol blended fuels. **Applied Thermal Engineering**, Elsevier, v. 95, p. 186–203, 2016. Citado na página 32.

NAZOKTABAR, Mohsen et al. Developing a multi-zone model for a hcci engine to obtain optimal conditions using genetic algorithm. **Energy conversion and management**, Elsevier, v. 157, p. 49–58, 2018. Citado na página 30.

NELLES, Oliver. **Nonlinear system identification: from classical approaches to neural networks and fuzzy models**. [S.l.]: Springer Science & Business Media, 2013. Citado na página 39.

NICKMEHR, Neda. **System Identification of an Engine-Load Setup Using Grey-Box Model**. 2014. Tese (Doutorado) — Linköping University Electronic Press, 2014. Citado na página 29.

NICOLAO, G De; SCATTOLINI, Riccardo; SIVIERO, C. Modelling the volumetric efficiency of ic engines: parametric, non-parametric and neural techniques. **Control Engineering Practice**, Elsevier, v. 4, n. 10, p. 1405–1415, 1996. Citado na página 31.

NIU, Xiaoxiao et al. Investigation of ann and svm based on limited samples for performance and emissions prediction of a crdi-assisted marine diesel engine. **Applied Thermal Engineering**, Elsevier, v. 111, p. 1353–1364, 2017. Citado na página 32.

NOLA, Francesco de et al. Volumetric efficiency estimation based on neural networks to reduce the experimental effort in engine base calibration. **Fuel**, Elsevier, v. 244, p. 31–39, 2019. Citado na página 31.

NOROUZI, Armin; ALIRAMEZANI, Masoud; KOCH, Charles Robert. A correlation-based model order reduction approach for a diesel engine nox and brake mean effective pressure dynamic model using machine learning. **International Journal of Engine Research**, SAGE Publications Sage UK: London, England, p. 1468087420936949, 2020. Citado na página 32.

OJHA, Varun Kumar; ABRAHAM, Ajith; SNÁŠEL, Václav. Metaheuristic design of feedforward neural networks: A review of two decades of research. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 60, p. 97–116, 2017. Citado na página 24.

PHAM, Duc Truong; CASTELLANI, Michele. The bees algorithm: modelling foraging behaviour to solve continuous optimization problems. **Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science**, SAGE Publications Sage UK: London, England, v. 223, n. 12, p. 2919–2938, 2009. Citado 2 vezes nas páginas 32 e 33.

PULKRABEK, Willard W. **Engineering fundamentals of the internal combustion engine**. 2004. Citado na página 18.

RAMADHAS, AS; JAYARAJ, S; MURALEEDHARAN, C. Theoretical modeling and experimental studies on biodiesel-fueled engine. **Renewable energy**, Elsevier, v. 31, n. 11, p. 1813–1826, 2006. Citado na página 28.

RÍOS, Jorge Enrique Gaviria; GUZMÁN, Jorge Hernán Mora; SANTAMARÍA, John Ramiro Agudelo. Historia de los motores de combustión interna. Universidad de Antioquia, Facultad de Ingeniería, 2002. Citado na página 18.

ROEPKE, Karsten. Design of experiments for engine calibration. **Journal of The Society of Instrument and Control Engineers**, The Society of Instrument and Control Engineers, v. 53, n. 4, p. 322–327, 2014. Citado na página 23.

ROJAS, Raul. The backpropagation algorithm. In: **Neural networks**. [S.l.]: Springer, 1996. p. 149–182. Citado na página 37.

SHAMEKHI, Amir-Mohammad; SHAMEKHI, Amir H. A new approach in improvement of mean value models for spark ignition engines using neural networks. **Expert Systems with Applications**, Elsevier, v. 42, n. 12, p. 5192–5218, 2015. Citado na página 29.

SIQUEIRA, Hugo et al. Performance analysis of unorganized machines in streamflow forecasting of brazilian plants. **Applied Soft Computing**, Elsevier, v. 68, p. 494–506, 2018. Citado na página 36.

SUBANA, Shanmuganathan; SAMARASINGHE, Sandhya. Artificial neural network modelling. **Ed. Springer, Cham**, 2016. Citado na página 33.

TILLMANN, Carlos Antonio da Costa. Motores de combustão interna e seus sistemas. **Pelotas: Instituto Federal de Educação, Ciência e Tecnologia**, p. 9–25, 2013. Citado 2 vezes nas páginas 18 e 19.

TURKSON, Richard Fiifi et al. Artificial neural network applications in the calibration of spark-ignition engines: An overview. **Engineering science and technology, an international journal**, Elsevier, v. 19, n. 3, p. 1346–1359, 2016. Citado 2 vezes nas páginas 23 e 24.

WAHAB, Mohd Nadhir Ab; NEFTI-MEZIANI, Samia; ATYABI, Adham. A comprehensive review of swarm optimization algorithms. **PloS one**, Public Library of Science, v. 10, n. 5, p. e0122827, 2015. Citado 2 vezes nas páginas 32 e 33.

WANG, Kang-Ping et al. Particle swarm optimization for traveling salesman problem. In: IEEE. **Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE cat. no. 03ex693)**. [S.l.], 2003. v. 3, p. 1583–1585. Citado 2 vezes nas páginas 32 e 33.

WANG, SW et al. Adaptive neural network model based predictive control for air–fuel ratio of si engines. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 19, n. 2, p. 189–200, 2006. Citado na página 15.

WHITMARSH, Lorraine; KÖHLER, Jonathan. Climate change and cars in the eu: the roles of auto firms, consumers, and policy in responding to global environmental change. **Cambridge Journal of Regions, Economy and Society**, Oxford University Press, v. 3, n. 3, p. 427–441, 2010. Citado na página 14.

YANG, Xin-She. **Engineering optimization: an introduction with metaheuristic applications**. [S.l.]: John Wiley & Sons, 2010. Citado na página 32.

_____. A new metaheuristic bat-inspired algorithm. In: **Nature inspired cooperative strategies for optimization (NICSO 2010)**. [S.l.]: Springer, 2010. p. 65–74. Citado 2 vezes nas páginas 32 e 33.

_____. Flower pollination algorithm for global optimization. In: SPRINGER. **International conference on unconventional computing and natural computation**. [S.l.], 2012. p. 240–249. Citado 2 vezes nas páginas 32 e 33.

YANG, Xin-She; DEB, Suash. Cuckoo search via lévy flights. In: IEEE. **2009 World congress on nature & biologically inspired computing (NaBIC)**. [S.l.], 2009. p. 210–214. Citado 2 vezes nas páginas 32 e 33.

ZANARDO, Gabriele et al. **Grey box control oriented SCR model**. [S.l.], 2013. Citado na página 29.

ZHAO, Jinxing et al. Design and optimization of an atkinson cycle engine with the artificial neural network method. **Applied energy**, Elsevier, v. 92, p. 492–502, 2012. Citado 2 vezes nas páginas 23 e 24.

ZHOU, Qiong et al. Neural network–based modeling and optimization for effective vehicle emission testing and engine calibration. **Chemical Engineering Communications**, Taylor & Francis, v. 195, n. 6, p. 706–720, 2008. Citado 2 vezes nas páginas 23 e 24.

ZOU, Jinming; HAN, Yi; SO, Sung-Sau. Overview of artificial neural networks. In: **Artificial Neural Networks**. [S.l.]: Springer, 2008. p. 14–22. Citado na página 25.