



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS  
DEPARTAMENTO DE ENGENHARIA MECÂNICA  
CURSO DE ENGENHARIA MECÂNICA

EDGLEYSON PEREIRA DA SILVA

**APLICAÇÃO DE TÉCNICAS DE REDES NEURAIS NA SIMULAÇÃO DE  
ESCOAMENTO DE GÁS NATURAL ENRIQUECIDO POR HIDROGÊNIO VERDE**

Recife  
2023

EDGLEYSON PEREIRA DA SILVA

**APLICAÇÃO DE TÉCNICAS DE REDES NEURAS NA SIMULAÇÃO DE  
ESCOAMENTO DE GÁS NATURAL ENRIQUECIDO POR HIDROGÊNIO VERDE**

Trabalho de Conclusão de Curso  
apresentado ao Curso de Engenharia  
Mecânica da Universidade Federal de  
Pernambuco, como requisito parcial para  
obtenção do título de Engenheiro  
Mecânico.

Orientador: Darlan Karlo Elisiario de Carvalho

Recife

2023

Ficha de identificação da obra elaborada pelo autor,  
através do programa de geração automática do SIB/UFPE

Silva, Edgleyson Pereira da.

Aplicação de técnicas de redes neurais na simulação de escoamento de gás natural enriquecido por hidrogênio verde / Edgleyson Pereira da Silva. - Recife, 2023.

51 p. : il., tab.

Orientador(a): Darlan Karlo Elisiario de Carvalho

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia Mecânica - Bacharelado, 2023.

1. Hidrogênio verde. 2. Rede neural profunda. 3. Aprendizado de máquina. 4. Simulação de dinâmica dos fluidos computacional. I. Carvalho, Darlan Karlo Elisiario de. (Orientação). II. Título.

620 CDD (22.ed.)

EDGLEYSON PEREIRA DA SILVA

**APLICAÇÃO DE TÉCNICAS DE REDES NEURAIIS NA SIMULAÇÃO DE  
ESCOAMENTO DE GÁS NATURAL ENRIQUECIDO POR HIDROGÊNIO VERDE**

Trabalho de Conclusão de Curso  
apresentado ao Curso de Engenharia  
Mecânica da Universidade Federal de  
Pernambuco, como requisito parcial para  
obtenção do título de Engenheiro  
Mecânico.

Aprovado em: 29/09/2023

**BANCA EXAMINADORA**

---

Profa. Dr. : Darlan Karlo Elisiário de Carvalho (Orientador)  
Universidade Federal de Pernambuco

---

Prof. Dr. Francisco Fernando Roberto Pereira (Examinador Interno)  
Universidade Federal de Pernambuco

---

Prof. Alessandro Romário Echevarria Antunes (Examinador Externo)  
Universidade Federal de Pernambuco

## RESUMO

O uso de hidrogênio verde em aplicações da engenharia vem crescendo como aproveitar esse recurso sustentável para descarbonizar setores-chave da economia e alcançar metas ambiciosas de redução de emissões de carbono. Uma das áreas de implementação seria em escoamento de gás natural, onde o enriquecimento por hidrogênio verde pode melhorar a eficiência do transporte de gás natural. Aliado a isso, a simulação de Dinâmica dos Fluidos Computacional mostra-se uma ferramenta valiosa, mas demorada, para investigar essa uniformidade devido a grande demanda computacional. Para sanar essa deficiência, o Modelo de Rede Neural Profunda apresenta-se como uma solução para acelerar a previsão da uniformidade da mistura sob várias condições de trabalho devido à sua capacidade de aprendizado e generalização a partir de dados. O objetivo deste trabalho é replicar um estudo de injeção de hidrogênio verde em um duto de gás natural e analisar como redes neurais podem ser usadas para entender o comportamento dos fluxos. Serão implementadas redes neurais eficientes, seguidas de validação e avaliação da abordagem. Para a rede foram necessários alguns parâmetros para serem os dados de entrada do treinamento, como a razão de mistura de hidrogênio, velocidade do gás, temperatura, diâmetro da tubulação entre outras características. Enquanto para sua saída, para melhor saber a condição da mistura dos gases, o coeficiente de variação de hidrogênio é o resultado para qual a rede deve ser treinada. Diversas simulações foram realizadas para verificação de resultados para testar a capacidade de generalização da rede em diferentes cenários operacionais. A análise dos resultados revelou discrepâncias entre os valores previstos pela rede e os valores calculados a partir da simulação. Essas diferenças foram exploradas em detalhes, considerando a qualidade dos dados de treinamento, a sensibilidade das características dos dados de entrada e a escolha dos hiperparâmetros do modelo.

**Palavras-chave:** Hidrogênio verde, rede neural profunda, aprendizado de máquina, simulação de dinâmica dos fluidos computacional.

## ABSTRACT

The use of green hydrogen in engineering applications is growing as a means to harness this sustainable resource for decarbonizing key sectors of the economy and achieving ambitious carbon emission reduction goals. One area of implementation would be in natural gas flow, where the enrichment with green hydrogen can improve the efficiency of natural gas transportation. In addition to this, Computational Fluid Dynamics (CFD) simulation proves to be a valuable but time-consuming tool for investigating this uniformity due to its significant computational demand. To address this deficiency, the Deep Neural Network Model presents itself as a solution to accelerate the prediction of mixture uniformity under various working conditions due to its learning and generalization capabilities from data. The aim of this work is to replicate a study of green hydrogen injection into a natural gas pipeline and analyze how neural networks can be used to understand flow behavior. Efficient neural networks will be implemented, followed by validation and assessment of the approach. Several parameters were required for the network to be used as input data for training, such as hydrogen blending ratio, gas velocity, temperature, pipe diameter, among other features. Meanwhile, for its output, to better understand the gas mixture condition, the coefficient of hydrogen variation is the result for which the network should be trained. Multiple simulations were conducted to verify results and test the network's generalization capability in different operational scenarios. The analysis of the results revealed discrepancies between the values predicted by the network and the values calculated from the simulation. These differences were explored in detail, considering the quality of the training data, the sensitivity of the input data features, and the choice of model hyperparameters.

**Keywords:** Green hydrogen, deep neural network, machine learning, computational fluid dynamics simulation.

## LISTA DE FIGURAS

Figura 1 – Algoritmos de Aprendizagem . . . . .	10
Figura 2 – Neurônio Natural . . . . .	11
Figura 3 – Neurônio Artificial . . . . .	12
Figura 4 – Função Sigmóide . . . . .	15
Figura 5 – Função ReLU . . . . .	16
Figura 6 – Função Tanh . . . . .	17
Figura 7 – Malha de Elementos Finitos . . . . .	22
Figura 8 – Malha Estruturada . . . . .	23
Figura 9 – Malha Não Estruturada . . . . .	23
Figura 10 – Metodologia de Pesquisa . . . . .	27
Figura 11 – Geometria da Simulação . . . . .	28
Figura 12 – Fração Molar na Seção de 1m do Tubo. . . . .	35
Figura 13 – Treinamento da Rede . . . . .	37
Figura 14 – Diferença do valor real pelo valor previsto na rede . . . . .	40

## LISTA DE TABELAS

Tabela 1 – Características do Duto . . . . .	28
Tabela 2 – Configurações do Modelo . . . . .	32
Tabela 3 – Dados de Entrada da Rede Neural . . . . .	34
Tabela 4 – Resultados da Predição . . . . .	39

## LISTA DE ABREVIATURAS E SIGLAS

Adam	Adaptive Moment Estimation
AF	Função de Ativação
Agrad	Adaptive Gradient Algorithm
CFD	Dinâmica dos Fluidos Computacional
COV	Coefficiente de Variação
DNN	Rede Neural Profunda
GN	Gás Natural
RNA	Rede Neural Artificial
SGD	Stochastic Gradient Descent
tanh	Tangente Hiperbólica

## LISTA DE SÍMBOLOS

$y$	Valor e saída do neurônio
$w$	Peso sináptico
$b$	Bias
$w^{(t)}$	Vetor peso
$\alpha$	Taxa de aprendizagem
$\theta$	Vetor de parâmetro
$m^{\wedge}t$	Média móvel dos gradientes de primeira ordem
$v^{\wedge}t$	Média móvel dos gradientes de segunda ordem
$\epsilon$	Fator que evita divisão por zero
$\eta$	Taxa de aprendizado
$G_t$	Soma dos quadrados dos gradientes
$\rho$	Densidade
$v$	Velocidade
$p$	Pressão
$\mu$	Viscosidade
$Y_i$	Espécie química
$V_i$	Velocidade molar
$D$	Coefficiente de difusão
$R_i$	Taxa de geração
$C_p$	Calor específico a pressão constante
$T$	Temperatura
$k$	Condutividade térmica
$Q$	Taxa de calor gerado
$u_i$	Componente de velocidade
$f_i$	Força do corpo gravitacional que atua sobre o fluxo de gás
$c$	Concentração volumétrica
$S$	Termo fonte

$G_k$	Geração de energia cinética turbulenta pelos gradientes de velocidade
$G_b$	Geração de energia cinética turbulenta devido à flutuabilidade
$Y_M$	Dilatação flutuante na turbulência compressível
$c_i$	Fração volumétrica no ponto de amostragem
$\bar{c}$	Fração volumétrica média em todos os pontos de amostragem
$n_i$	Número de mols no trecho
$n_t$	Número de mols total da mistura
$V_i$	Volume do componente
$V_{trecho}$	Volume total do trecho da mistura
$R$	Constante dos gases ideais

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>7</b>
1.1	OBJETIVO GERAL	8
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>9</b>
2.1	REDES NEURAI E APRENDIZAGEM PROFUNDA	9
<b>2.1.1</b>	<b>Algoritmos de aprendizagem</b>	<b>9</b>
<b>2.1.2</b>	<b>Visão Geral das Redes Neurais Artificiais</b>	<b>11</b>
<b>2.1.3</b>	<b>A Arquitetura de uma Rede Neural</b>	<b>12</b>
2.1.3.1	Feedforward e backpropagation	13
<b>2.1.4</b>	<b>Funções de Ativação</b>	<b>14</b>
2.1.4.1	Função sigmóide	14
2.1.4.2	Função ReLU	15
2.1.4.3	Função TanH	17
<b>2.1.5</b>	<b>Algoritmos de Otimização</b>	<b>18</b>
2.1.5.1	Stochastic Gradient Descent	18
2.1.5.2	Adaptive Moment Estimation	19
2.1.5.3	Adaptive Gradient Algorithm	19
2.2	DINÂMICA DOS FLUIDOS COMPUTACIONAL	20
<b>2.2.1</b>	<b>Conceitos de Dinâmica dos Fluidos Computacional</b>	<b>20</b>
2.3	ANÁLISE COMPUTACIONAL	21
<b>2.3.1</b>	<b>Geração de Malha</b>	<b>21</b>
<b>2.3.2</b>	<b>Modelos de CFD</b>	<b>23</b>
<b>2.3.3</b>	<b>Visão Geral do Software</b>	<b>24</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>26</b>
3.1	PROCEDIMENTOS DE COLETA DE DADOS	26
3.2	CARACTERÍSTICAS DO MODELO	27
<b>3.2.1</b>	<b>Geometria de Simulação</b>	<b>27</b>
<b>3.2.2</b>	<b>Matemática da Simulação</b>	<b>28</b>
<b>3.2.3</b>	<b>Geração de Malha</b>	<b>30</b>
<b>3.2.4</b>	<b>Método de Solução da Simulação CFD</b>	<b>30</b>
3.3	MODELO DE REDE NEURAL PROFUNDA	32
<b>3.3.1</b>	<b>Google Colab</b>	<b>33</b>
<b>3.3.2</b>	<b>Previsão do ??</b>	<b>33</b>
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>35</b>
4.1	DETERMINAÇÃO DO COV	35
4.2	TREINAMENTO DA REDE	37
<b>4.2.1</b>	<b>Verificação de Resultados</b>	<b>38</b>
<b>4.2.2</b>	<b>Análise dos Resultados</b>	<b>38</b>

5	CONCLUSÃO . . . . .	41
6	TRABALHOS FUTUROS . . . . .	42
7	REFERÊNCIAS . . . . .	43

## 1 INTRODUÇÃO

O Brasil é um dos maiores emissores de dióxido de carbono ( $CO_2$ ) no mundo, ocasionando significativas contribuições para o aquecimento global. A inclusão do hidrogênio verde ( $H_2V$ ) na rede de GN pode ser uma solução para a descarbonização da matriz energética brasileira, mas, para que isso seja possível, é necessário a criação de regulamentações técnicas para sua produção e distribuição deste combustível (NUNES *et al.*, 2017). O hidrogênio verde ( $H_2V$ ) é amplamente reconhecido como o combustível mais limpo do mundo e é apontado como uma das fontes de energia mais sustentáveis e importantes para o futuro (Liu *et al.*, 2019).

O  $H_2V$  é produzido a partir de fontes renováveis de energia, como a eólica e a solar, e sua queima produz apenas água como resíduo, o que o torna uma opção atrativa para a descarbonização da matriz energética. A presente pesquisa propõe a aplicação de técnicas de redes neurais artificiais para a predição de características do escoamento da mistura de gás natural enriquecido por hidrogênio, utilizando informações obtidas através da simulação numérica computacional usando CFD deste escoamento como banco de dados para treinar e validar a rede, e assim tornar possível a obtenção das referidas características de forma mais rápida que esta simulação.

Kochkov *et al.* (2021) abordam que a simulação numérica é uma técnica poderosa para prever o comportamento de fluidos em sistemas complexos, mas pode ser muito demorada e exigir recursos computacionais caros para executar. Ao treinar redes neurais artificiais usando dados obtidos a partir de simulações numéricas, é possível acelerar o processo de simulação e reduzir o custo computacional. Os modelos de redes neurais artificiais são capazes de prever o comportamento do fluido com alta precisão em uma fração do tempo necessário para executar simulações numéricas completas. Além disso, os modelos de redes neurais podem ser treinados usando dados de simulações numéricas em condições de fluxo extremas, que não podem ser facilmente alcançadas na prática. Isso permite que os modelos sejam usados para prever o comportamento do fluido em situações extremas e otimizar o projeto de sistemas complexos. Portanto, o uso de *machine learning* em conjunto com simulações numéricas é justificado como uma forma de acelerar o processo de simulação e melhorar a precisão das previsões de comportamento do fluido em sistemas complexos. Além disso, a implementação de redes neurais na simulação permite acelerar a obtenção de resultados.

Como visto no trabalho de Su *et al.* (2022), as simulações de injeção de hidrogênio verde nas redes de gás natural envolvem muitos sistemas físicos complexos descritos por equações diferenciais parciais não lineares, equações de movimento e leis básicas da física. Para avaliar a dinâmica dos fluidos, as equações de Navier-Stokes são as mais adequadas, no entanto, resolvê-las em escala real é uma tarefa desafiadora devido ao custo computacional elevado, acarretando a uma demora na obtenção dos resultados precisos. A integração de

técnicas de aprendizagem profunda e redes neurais na simulação de Dinâmica dos Fluidos Computacional busca melhorar a eficiência da modelagem dos gasodutos, permitindo uma avaliação mais rápida e eficiente das características do escoamento analisado, tornando-se uma ferramenta valiosa para o estudo da viabilidade da injeção de hidrogênio verde na rede de gás natural. O objetivo final é otimizar o tempo de análise de características da injeção de hidrogênio verde nas redes de gás natural, com resultados precisos e rápidos.

Após a criação e estruturação da rede, foram conduzidos testes para verificar sua eficácia. À medida que os resultados eram comparados, observou-se que alguns deles apresentavam discrepâncias em relação aos resultados da simulação ou até mesmo pareciam não fazer sentido fisicamente. Diante dessa observação, uma análise minuciosa dos possíveis fatores que contribuíram para essas diferenças foi realizada, visando identificar oportunidades de aprimoramento para trabalhos futuros.

## 1.1 OBJETIVO GERAL

Aplicar técnicas de redes neurais na simulação do transporte da otimizar o tempo de análise de características do transporte de hidrogênio verde nas redes de gás natural, com resultados precisos e rápidos. Para tanto foram estabelecidos como objetivos específicos:

- Realizar o estudo do caso de injeção de hidrogênio verde em um duto de gás natural;
- Analisar a aplicação de técnicas de aprendizagem profunda e redes neurais em problemas de análise do comportamento de escoamentos;
- Aplicar as bibliotecas disponíveis para implementar redes neurais;
- Implementar redes neurais capazes de generalizar, com eficiência e baixo custo computacional, características do escoamento da mistura de hidrogênio verde e gás natural;
- Realizar testes de validação do código implementado;
- Avaliar a eficácia da abordagem adotada.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, apresenta-se uma revisão bibliográfica sobre técnicas de aprendizagem profunda, redes neurais e simulação CFD. São abordados temas como algoritmos de aprendizagem, visão geral das redes neurais artificiais, estrutura e funções das camadas da rede, funções de ativação, retropropagação e algoritmos de otimização, além de equações governantes da dinâmica dos fluidos computacional.

### 2.1 REDES NEURAI E APRENDIZAGEM PROFUNDA

Redes neurais são amplamente utilizados na simulação numérica de uma variedade de fenômenos físicos, incluindo problemas de dinâmica dos fluidos, transferência de calor, transferência de massa e outros processos de transporte. Eles oferecem a vantagem de representar fisicamente a conservação das quantidades físicas, tornando-os uma abordagem poderosa para a modelagem e análise de sistemas complexos em engenharia, ciências naturais e muitas outras áreas.

#### 2.1.1 Algoritmos de aprendizagem

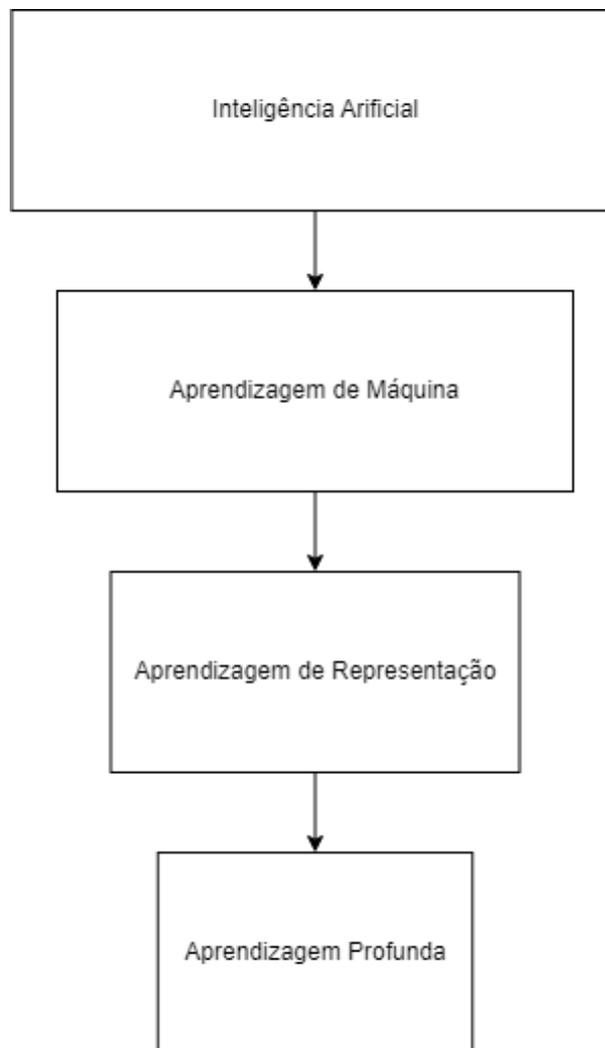
Os algoritmos de aprendizagem, Figura 1, são capazes de aprender com dados para melhorar o desempenho em uma tarefa específica medida por uma métrica de desempenho. As experiências, tarefas e medidas de desempenho variam muito e não há uma definição formal para cada uma dessas entidades. A aprendizagem de máquina nos permite lidar com tarefas que seriam muito difíceis de serem resolvidas por meio de programas fixos. As tarefas são descritas em termos de como o sistema deve processar um exemplo, que é uma coleção de recursos quantitativos medidos a partir de algum objeto ou evento. O objetivo é alcançar a tarefa de aprendizagem, enquanto a aprendizagem é o meio pelo qual esse objetivo é alcançado (MITCHELL, 1997).

De acordo Goodfellow *et al.* (2010), muitos tipos de tarefas podem ser resolvidos com a aprendizagem de máquina. Dentre eles há a classificação, em que o programa de computador é solicitado a especificar em qual das  $k$  categorias uma entrada pertence. Para resolver essa tarefa, o algoritmo de aprendizado é, geralmente, solicitado a produzir uma função que atribui a entrada descrita a uma categoria identificada pelo código numérico. Outro tipo de tarefa é a regressão, onde o programa de computador é solicitado a prever um valor numérico, dada uma entrada. Essa tarefa é semelhante à classificação, exceto que o formato da saída é diferente.

Dentre os algoritmos de aprendizado abordados, a aprendizagem profunda é uma subárea da aprendizagem de máquina que se concentra em redes neurais profundas com muitas camadas, capazes de aprender representações complexas dos dados. Há também a aprendizagem de representação que é uma técnica utilizada na aprendizagem de máquina

para ensinar sistemas de computadores a identificar características importantes dos dados e transformá-las em representações mais simples. Já a aprendizagem de máquina é uma subárea da inteligência artificial que utiliza algoritmos e modelos estatísticos para permitir que os sistemas de computadores aprendam com os dados e melhorem seu desempenho em uma determinada tarefa. Pode ser visto na Figura 1 um fluxograma que mostra como a aprendizagem profunda é um tipo de aprendizagem de representação, que por sua vez é um tipo de aprendizagem de máquina, que é usado para muitas, mas não todas, em abordagens de inteligência artificial. E cada seção do diagrama de Venn inclui um exemplo de uma tecnologia de inteligência artificial.

Figura 1 – Algoritmos de Aprendizagem



Fonte: Autor.

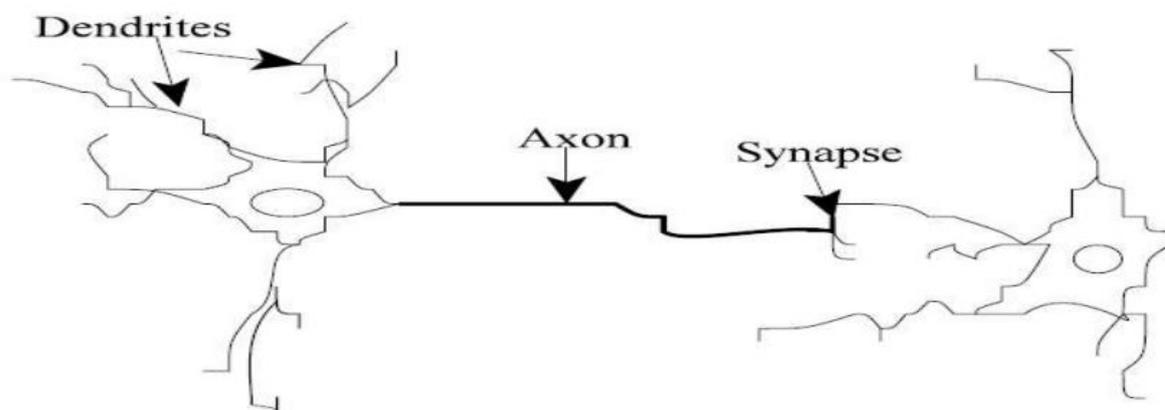
Além disso, os algoritmos de aprendizagem de máquinas podem ser amplamente categorizados como não supervisionados ou supervisionados pelo tipo de experiência que lhes é permitido ter durante o processo de aprendizagem. Algoritmos de aprendizado não supervisionado aprendem propriedades úteis da estrutura do conjunto de dados, sem a necessidade de rótulos ou metas. Já em aprendizado supervisionado, cada exemplo é

associado a um rótulo ou meta. Em geral, em aprendizado profundo, busca-se aprender toda a distribuição de probabilidade que gerou o conjunto de dados, seja de forma explícita, como em estimativa de densidade, ou implicitamente, para tarefas como síntese ou remoção de ruído. Outros algoritmos de aprendizado não supervisionado executam outras tarefas, como agrupamento, que consiste em dividir o conjunto de dados em grupos de exemplos semelhantes.

### 2.1.2 Visão Geral das Redes Neurais Artificiais

O sucesso da Inteligência Artificial se deve em grande parte ao aprendizado profundo, que usa redes neurais artificiais como modelo subjacente. Embora vagamente baseadas em redes neurais biológicas, vistas na Figura 2, as redes neurais artificiais são provavelmente a melhor maneira de especificar um conjunto flexível de funções, construídas a partir de muitos blocos computacionais básicos chamados neurônios. O poder real do aprendizado profundo vem das redes neurais profundas, com muitos neurônios em paralelo organizados em camadas computacionais sequenciais, aprendendo representações úteis do mundo (ROBERTS *et al.*, 2022).

Figura 2 – Neurônio Natural



Fonte: Zakaria; Mabrouka; Sarhan (2014).

Esses modelos são baseados na teoria da otimização e possuem uma estrutura semelhante ao funcionamento do cérebro humano. Eles são compostos por neurônios artificiais, também conhecidos como unidades de processamento, que estão interligados entre si por meio de pesos que representam as conexões entre os neurônios, como visto na Figura 3. Esses neurônios são organizados em camadas e o termo "rede" refere-se à interconexão entre eles, que determina o impacto de um neurônio sobre outro. O peso da rede neural é crucial para o funcionamento desses modelos, pois determina como os neurônios são interligados e como a rede processa as informações de entrada para gerar uma saída (ZAKARIA; MABROUKA; SARHAN, 2014).

Figura 3 – Neurônio Artificial



Fonte: Autor.

### 2.1.3 A Arquitetura de uma Rede Neural

Como explica Aggarwal (2018), uma rede neural de camada única é uma instância simples de uma rede neural em que as entradas são diretamente mapeadas para uma única saída através de uma variação generalizada de uma função linear. Essa instância é chamada de *perceptron*. Por outro lado, as redes neurais multicamadas possuem neurônios organizados em camadas, com as camadas de entrada e saída sendo separadas por camadas ocultas. Essa arquitetura de rede neural é conhecida como rede de alimentação. As camadas ocultas permitem que a rede neural seja capaz de aprender representações mais complexas e abstratas dos dados de entrada.

Goodfellow *et al.*, (2010) também explicam que a arquitetura de uma rede neural é a definição da estrutura geral da rede, incluindo o número de unidades ou neurônios que compõem a rede e como essas unidades estão interconectadas. As redes neurais são geralmente organizadas em camadas, que são grupos de unidades que processam informações de forma hierárquica. Cada camada pode ser considerada uma função da camada anterior, ou seja, os dados de entrada são processados por uma camada e passam para a próxima camada para extração de características mais complexas. Essa organização em cadeia é comum em muitas arquiteturas de redes neurais, como as redes neurais *feedforward*, em que a informação flui em uma direção, da entrada para a saída. Cada camada da rede neural desempenha um papel específico na transformação dos dados de entrada em saída, permitindo que a rede aprenda representações cada vez mais abstratas e complexas à medida que a informação é processada nas camadas subsequentes.

As entradas são propagadas por meio da topologia da RNA, passando a se tornar pesos sinápticos, que seriam as forças de conexão entre neurônios, e por uma *activation function* (AF) ou função de ativação dos neurônios, que é uma função matemática que determina a saída de um neurônio com base em sua entrada (MACHADO e FONSECA JÚNIOR, 2015). Recebendo entradas de  $n$  neurônios ( $y_i$ ) o neurônio  $k$  calcula a saída  $y_k$  através da Equação (1).

$$y_k = AF \left( \sum_{i=1}^n (y_i w_{ki} + b_k) \right) \quad (1)$$

Da Equação (1),  $y_i$  trata-se da saída calculada pelo neurônio  $i$ ,  $w_{ki}$  é o peso sináptico entre o neurônio  $i$  e o neurônio  $k$  e  $b_k$  o peso entre um valor que é constante e diferente de zero ao neurônio  $k$ , denominado como *bias*. Ao neurônio estar ligado às entradas, o termo  $y_i$  é então substituído pela entrada correspondente (MACHADO e FONSECA JÚNIOR, 2015).

Os parâmetros iniciais são definidos no início do treinamento e, a partir dos valores dos pesos sinápticos e *bias*, a rede é ajustada durante um conjunto de iterações, conhecidas como épocas, com base em um conjunto de dados de treinamento. Durante o treinamento, a RNA é alimentada com exemplos de entrada e a saída obtida pela rede é comparada com a saída desejada, também conhecida como rótulo. Para atualizar os pesos sinápticos e *bias*, são utilizadas técnicas de otimização, como o gradiente descendente, que ajustam os valores dos parâmetros com base na diferença entre a saída obtida e a saída desejada. Esse processo de atualização dos pesos e *bias* é repetido até que a rede alcance a convergência, ou seja, até que os parâmetros atinjam valores que levem a um desempenho adequado na tarefa de interesse (GOODFELLOW *et al.*, 2016).

Conforme destacado por Ruder (2016), o treinamento pode ser considerado completo quando a rede neural atinge um critério de parada previamente estabelecido, como o número máximo de épocas ou uma margem de erro aceitável. A determinação dos pesos sinápticos e *bias* é essencial para o correto funcionamento de uma RNA e é um dos principais aspectos a serem considerados na concepção e implementação de redes neurais em arquiteturas de redes complexas. A estimativa do *bias* e pesos sinápticos é tida como a fase de treinamento, e se constitui em um processo iterativo em que os parâmetros iniciais são utilizados até a convergência do processo.

### 2.1.3.1 Feedforward e backpropagation

Como dito anteriormente, as redes neurais artificiais são estruturas complexas que visam a modelagem matemática de problemas, a partir de uma abordagem inspirada no funcionamento do cérebro humano. Uma das arquiteturas mais comuns de redes neurais artificiais é a rede *feedforward*, também conhecida como rede de alimentação, na qual os sinais de entrada são propagados pela rede neural em uma única direção, da camada de entrada para a camada de saída, sem *feedback* ou conexões retroativas. O processo de propagação é realizado por meio da aplicação de pesos sinápticos, que são ajustados durante o treinamento da rede, e por AF, que são utilizadas para processar os sinais de entrada e gerar sinais de saída. O objetivo do treinamento é ajustar os pesos sinápticos para que a rede neural possa produzir saídas precisas e adequadas para uma dada entrada (GOODFELLOW *et al.*, 2016).

O algoritmo de *backpropagation* é uma técnica de aprendizagem supervisionada que é amplamente utilizada para ajustar os pesos sinápticos em redes neurais *feedforward*. O processo de *backpropagation* consiste em calcular o gradiente da função de erro em relação aos pesos sinápticos e, em seguida, usar esse gradiente para ajustar os pesos de modo a minimizar o erro da rede. Durante o processo de *backpropagation*, os sinais de erro são propagados da camada de saída para a camada de entrada da rede, e o gradiente da função de erro em relação aos pesos é calculado por meio da aplicação da regra da cadeia. Esse gradiente é usado para atualizar os pesos sinápticos de forma iterativa, até que a rede neural atinja a convergência, ou seja, até que os pesos atinjam valores que levem a um desempenho adequado na tarefa de interesse (RUMELHART *et al.*, 1986).

Bishop (2006) considera o *backpropagation* uma técnica eficiente de aprendizado de máquina, pois permite a adaptação da rede neural a diferentes tipos de dados e tarefas, desde que haja dados suficientes para treinamento. Além disso, a arquitetura de rede *feedforward* é capaz de aprender representações cada vez mais abstratas e complexas dos dados de entrada, à medida que a informação é processada nas camadas subsequentes, o que a torna uma ferramenta poderosa para diversas aplicações em áreas como reconhecimento de padrões, processamento de imagens, processamento de voz e muitas outras.

#### 2.1.4 Funções de Ativação

As funções de ativação são elementos-chave em redes neurais artificiais, responsáveis por introduzir a não-linearidade nas saídas dos neurônios, permitindo que as redes neurais aprendam a modelar relações complexas nos dados de entrada. Elas são aplicadas após a soma ponderada das entradas e dos pesos sinápticos em cada neurônio da rede. Quando ativadas, elas geram a saída do neurônio, que é transmitida para as camadas subsequentes da rede. (GOODFELLOW *et al.*, 2010).

As funções de ativação permitem que as redes neurais aprendam representações hierárquicas de características dos dados de entrada, possibilitando a extração de padrões e a tomada de decisões mais complexas. Goodfellow *et al.* (2010) destaca que as funções de ativação mais comuns são a sigmóide, *ReLU* e *tanh*. A sigmoide é uma função suave e contínua que mapeia os valores de entrada para um intervalo entre 0 e 1. A função *ReLU* é uma AF que zera os valores de entrada negativos e mantém os valores positivos inalterados. Já a função *tanh* é uma versão escalada da função sigmoide, mapeando os valores de entrada para um intervalo entre -1 e 1.

##### 2.1.4.1 Função sigmóide

A função sigmóide é uma função suave e contínua que mapeia os valores de entrada para um intervalo entre 0 e 1. Ela é caracterizada por uma saída próxima de 0 quando a entrada tende a menos infinito e próxima de 1 quando a entrada tende a mais infinito. A função sigmóide é amplamente usada em problemas de classificação binária, onde a rede

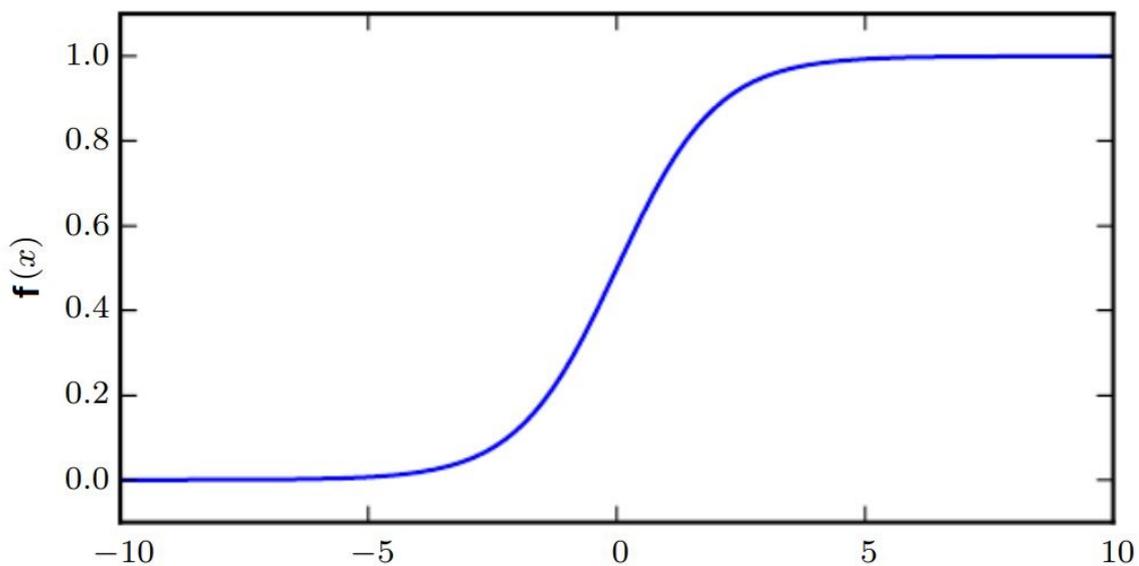
neural precisa atribuir uma saída binária (0 ou 1) a uma determinada entrada, pois mapeia a saída para uma probabilidade, onde 0 representa a probabilidade de pertencer à classe negativa e 1 representa a probabilidade de pertencer à classe positiva (GOODFELLOW *et al.*, 2010).

Ela é expressa pela Equação (2).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

Na Equação (2)  $e^{-x}$  é a função exponencial. A função sigmoide é caracterizada por uma saída suave e contínua, que se aproxima de 0 quando a entrada tende a menos infinito e se aproxima de 1 quando a entrada tende a mais infinito. Ela é amplamente usada em problemas de classificação binária, onde a rede neural precisa atribuir uma saída binária (0 ou 1) a uma determinada entrada (GOODFELLOW *et al.*, 2010). Ela pode ser vista na Figura 4.

Figura 4 – Função Sigmóide



Fonte: Goodfellow *et al.* (2010).

A função sigmoide possui uma desvantagem conhecida como *vanishing gradient*, que ocorre quando a derivada da função se torna muito pequena para valores de entrada distantes do zero, levando a uma desaceleração no processo de treinamento da rede neural. Além disso, a função sigmoide não é centrada em zero, o que pode afetar o aprendizado em redes neurais profundas (GÉRON, 2017).

#### 2.1.4.2 Função ReLU

A função *ReLU*, Figura 5, é uma AF popular que mapeia os valores de entrada para zero quando são negativos e mantém os valores positivos inalterados. A função

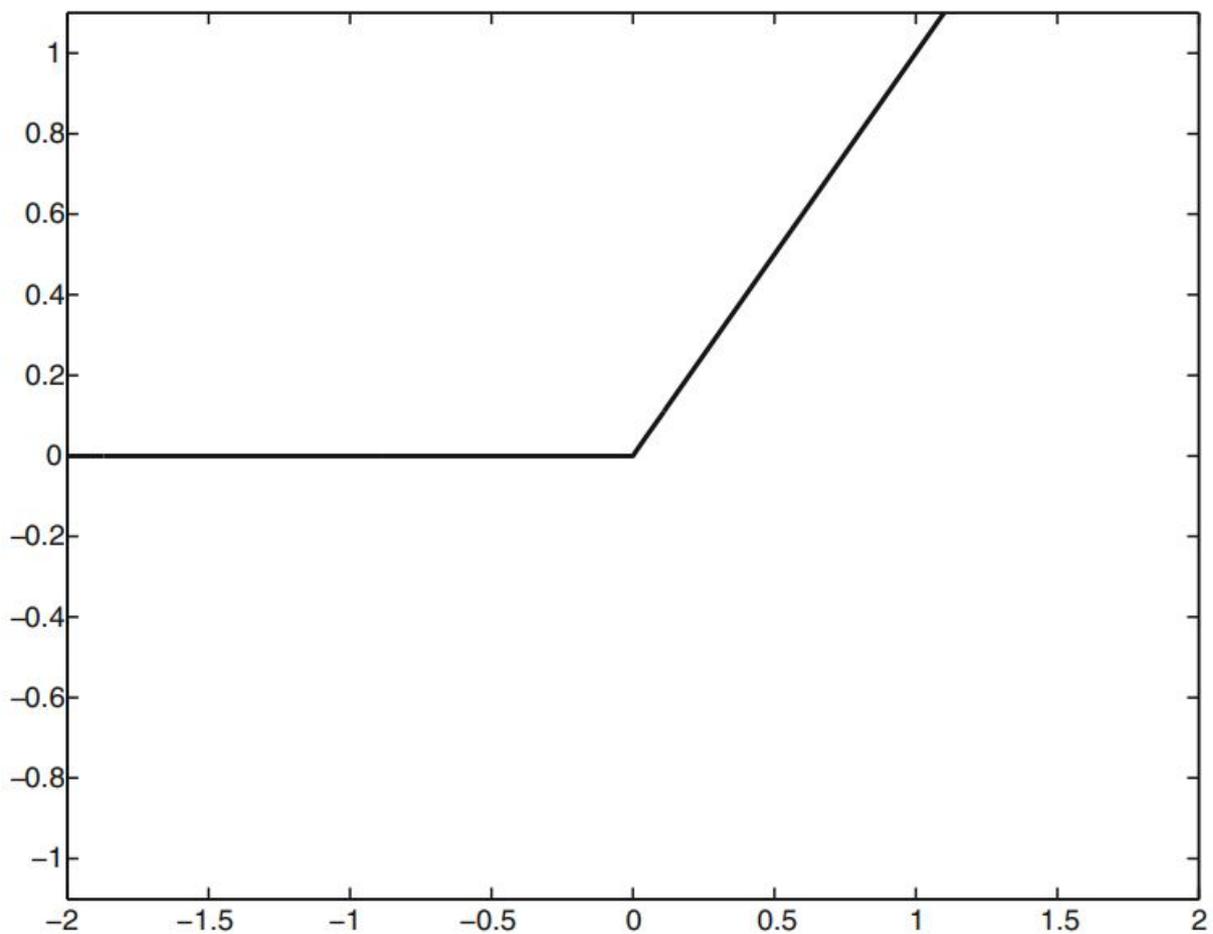
*ReLU* é caracterizada por ser computacionalmente eficiente, uma vez que apenas os valores negativos são zerados, o que acelera o processo de treinamento da rede neural (AGGARWAL, 2018).

Ela é expressa pela Figura 5.

$$f(x) = \max(0, x) \quad (3)$$

Na Figura 5 *max* é a função de máximo. E como explicado por Aggarwal (2018), a função *ReLU* é caracterizada por ser computacionalmente eficiente, uma vez que apenas os valores negativos são zerados, e tem a vantagem de mitigar o problema de gradiente desaparecente, que pode ocorrer em redes neurais profundas. Ela é frequentemente usada em muitas arquiteturas de redes neurais modernas, como a rede neural convolucional.

Figura 5 – Função ReLU



Fonte: Aggarwal (2018).

Além disso, a função *ReLU* resolve o problema de "*vanishing gradient*", uma vez que sua derivada é igual a 1 para valores de entrada positivos, o que evita o desaceleração do treinamento em redes neurais profundas. No entanto, a função *ReLU* possui uma desvantagem, que é a inatividade de alguns neurônios, uma vez que a saída é zero

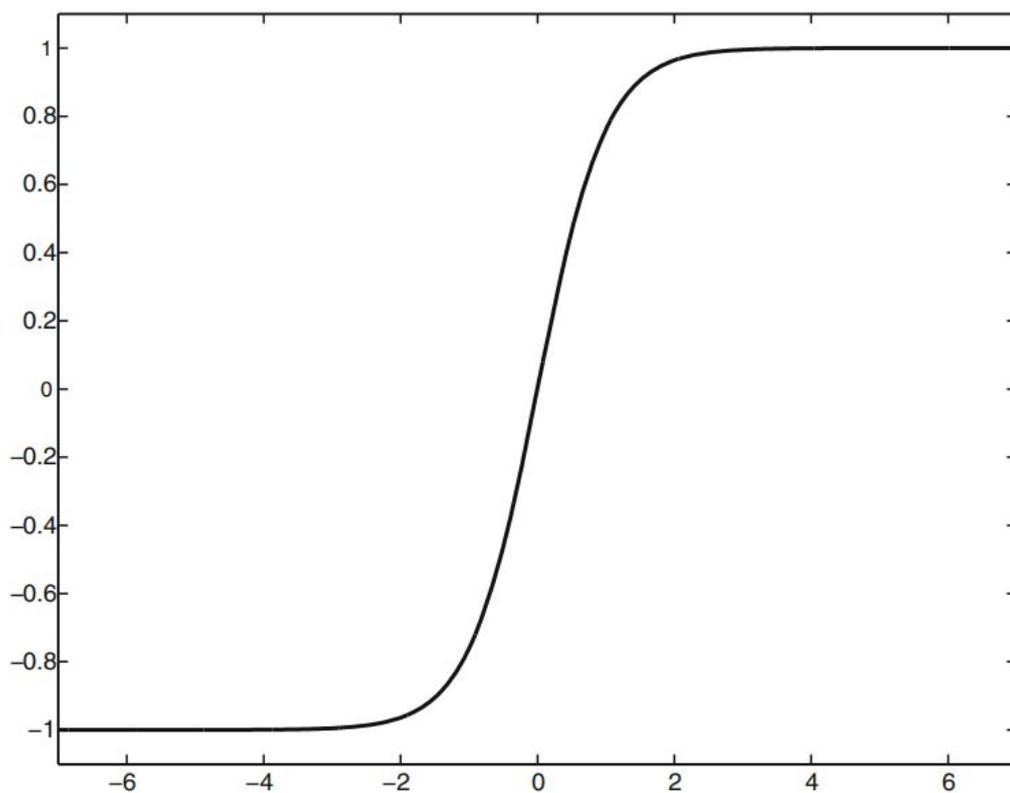
para valores negativos de entrada, o que pode levar à perda de informações importantes (GÉRON, 2017).

### 2.1.4.3 Função TanH

A AF tangente hiperbólica  $\tanh$  é uma das funções mais utilizadas em redes neurais devido às suas propriedades e capacidade de modelar não-linearidades. A função  $\tanh$  é uma função matemática que mapeia um número real para um intervalo entre -1 e 1. A função  $\tanh$  tem uma forma de S simétrica em torno do eixo  $y$ , o que significa que ela é positiva para valores de entrada positivos e negativa para valores de entrada negativos (GOODFELLOW *et al.*, 2016; RASCHKA e MIRJALILI, 2020), sendo definida pela Figura 6.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

Figura 6 – Função Tanh



Fonte: Aggarwal (2018).

A AF  $\tanh$  é usada para introduzir não-linearidade em redes neurais e é comumente aplicada em camadas ocultas. Ela é muito similar à função sigmóide, porém possui uma saída que varia entre -1 e 1, enquanto a sigmóide varia de 0 a 1. Isso significa que a função  $\tanh$  pode lidar com entradas positivas e negativas, enquanto a sigmóide funciona bem apenas com entradas positivas (AGGARWAL, 2018; BISHOP, 2006).

Em resumo, a AF *tanh* é uma função matemática que mapeia um valor de entrada em um intervalo entre -1 e 1, introduzindo não-linearidade em redes neurais. É uma função muito utilizada em camadas ocultas devido à sua capacidade de lidar com entradas positivas e negativas e sua similaridade com a função sigmóide (CHOLLET, 2018; GÉRON, 2019).

### 2.1.5 Algoritmos de Otimização

As redes neurais artificiais são estruturas complexas que visam a modelagem matemática de problemas, a partir de uma abordagem inspirada no funcionamento do cérebro humano. Uma das etapas mais importantes no uso de redes neurais é o treinamento da rede para que ela possa produzir saídas precisas e adequadas para uma dada entrada. Existem diversos algoritmos de treinamento que podem ser utilizados para ajustar os pesos sinápticos em uma rede neural (GOODFELLOW *et al.*, 2016).

Entre os algoritmos de otimização mais utilizados, destacam-se o *Stochastic Gradient Descent* (SGD), que é um método iterativo para otimizar uma função de custo, buscando os mínimos locais (BISHOP, 2006), e o *Adaptive Moment Estimation* (Adam), um algoritmo de otimização estocástica que calcula as médias móveis dos gradientes de primeira e segunda ordem, ajustando adaptativamente a taxa de aprendizado para cada parâmetro (KINGMA e BA, 2015). Outro algoritmo de treinamento amplamente utilizado é o *Adaptive Gradient Algorithm* (Agrad), que adapta a taxa de aprendizado de cada parâmetro com base nas suas histórias de gradiente (Duchi *et al.*, 2011). O uso adequado de cada um desses algoritmos é crucial para garantir a eficiência do treinamento e o desempenho da rede neural em tarefas específicas.

#### 2.1.5.1 Stochastic Gradient Descent

O algoritmo SGD é uma técnica amplamente utilizada para treinamento de redes neurais profundas. Ele é baseado no método de otimização *Gradient Descent* e é aprimorado pela introdução de uma abordagem estocástica para o cálculo do gradiente da função de perda. A ideia central do SGD é atualizar os pesos da rede de forma incremental, a partir da avaliação de um subconjunto aleatório de dados de treinamento, chamado de *mini-batch*. Essa abordagem reduz significativamente o custo computacional do treinamento e pode acelerar a convergência para uma solução ótima. A atualização dos pesos no SGD é dada pela Equação (5) (GOODFELLOW *et al.*, 2016).

$$w^{(t+1)} = w^{(t)} - \alpha \nabla_{w^{(t)}} L(w^{(t)}) \quad (5)$$

Onde  $w^{(t)}$  representa o vetor de pesos na iteração  $t$ ,  $\alpha$  é a taxa de aprendizagem que controla o tamanho do passo em cada atualização de peso,  $\nabla_{w^{(t)}} L(w^{(t)})$  é o gradiente da função de perda  $L$  em relação aos pesos  $w^{(t)}$  na iteração  $t$ . O gradiente é calculado

para um *mini-batch* aleatório de dados de treinamento a cada iteração. A atualização dos pesos é realizada de forma incremental, ou seja, a cada iteração, o novo vetor de pesos  $w^{(t+1)}$  é calculado a partir do vetor de pesos anterior  $w^{(t)}$ , subtraindo um múltiplo do gradiente da função de perda. O SGD repete esse processo até que a função de perda seja minimizada.

### 2.1.5.2 Adaptive Moment Estimation

O *Adam* (*Adaptive Moment Estimation*) é um algoritmo de otimização estocástica, que combina ideias do método de *Momentum* e do Agrad (GOODFELLOW *et al.*, 2016). O algoritmo é amplamente utilizado em redes neurais profundas para otimizar a função de custo durante o treinamento. O algoritmo de otimização *Adam* calcula as médias móveis dos gradientes de primeira e segunda ordem, ajustando adaptativamente a taxa de aprendizado para cada parâmetro. A atualização dos pesos é feita seguindo a Equação (6) (KINGMA e BA, 2015).

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t + \epsilon} \cdot \hat{m}_t} \quad (6)$$

Em que  $\theta_t$  é o vetor de parâmetros no passo  $t$ ,  $\alpha$  é a taxa de aprendizado,  $\hat{m}_t$  é a média móvel dos gradientes de primeira ordem e  $\hat{v}_t$  é a média móvel dos gradientes de segunda ordem. O parâmetro  $\epsilon$  é um pequeno número para evitar divisão por zero. O algoritmo de otimização *Adam* tem sido amplamente utilizado em diversas aplicações de aprendizagem profunda, sendo um dos algoritmos mais eficientes em termos de convergência e estabilidade do treinamento (AGGARWAL, 2018).

### 2.1.5.3 Adaptive Gradient Algorithm

O *Adaptive Gradient Algorithm* (Agrad) é um algoritmo de otimização que adapta a taxa de aprendizado de cada parâmetro com base em sua história de gradiente. O Agrad é um algoritmo iterativo que utiliza um esquema de aprendizado adaptativo para atualizar os pesos da rede. Ele tem como objetivo ajustar a taxa de aprendizagem de cada parâmetro de acordo com a frequência e magnitude de atualizações anteriores, a fim de melhorar a eficiência do processo de treinamento (DUCHI *et al.*, 2011).

A principal diferença do Agrad em relação ao *Gradient Descent* é que, em vez de usar uma taxa de aprendizado fixa, a taxa de aprendizado é ajustada para cada parâmetro durante o processo de treinamento. Isso permite que o Agrad se adapte automaticamente a diferentes taxas de aprendizado para diferentes parâmetros. Além disso, o Agrad é especialmente eficaz em problemas em que há uma grande variação na magnitude dos gradientes dos diferentes parâmetros. A atualização dos pesos no Agrad é dada pela Equação (7) (AGGARWAL, 2018).

$$w_{t+1,i} = w_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot \sum_{j=1}^t g_{j,i} \quad (7)$$

em que  $w_{t+1,i}$  é o vetor de pesos no instante  $t$  do  $i$ -ésimo parâmetro,  $g_{j,i}$  é o vetor gradiente do  $i$ -ésimo parâmetro no instante  $t$ ,  $G_{t,ii}$  é a soma dos quadrados dos gradientes anteriores do  $i$ -ésimo parâmetro até o instante  $t$ ,  $\eta$  é a taxa de aprendizado, e  $\epsilon$  é um parâmetro de suavização para evitar a divisão por zero.

## 2.2 DINÂMICA DOS FLUIDOS COMPUTACIONAL

### 2.2.1 Conceitos de Dinâmica dos Fluidos Computacional

Versteeg e Malalasekere (1995) explicam que a Dinâmica dos Fluidos Computacional CFD é uma técnica que envolve a simulação computadorizada de sistemas que apresentam fluxo de fluidos, transferência de calor e fenômenos associados, como reações químicas. Essa técnica é bastante poderosa e pode ser aplicada em uma ampla gama de áreas industriais e não industriais. Eles citam que entre as equações mais importantes da CFD, pode-se citar a equação de Navier-Stokes, Equação (8), que é fundamental para descrever o comportamento de fluidos viscosos. Essa equação leva em consideração a densidade do fluido  $\rho$ , sua velocidade  $v$ , a pressão  $p$ , a viscosidade dinâmica  $\mu$ , a aceleração da gravidade  $g$  e o operador gradiente  $\nabla$ .

$$\rho \left( \frac{dv}{dt} \right) = -\nabla p + \mu \nabla^2 v + \rho g \quad (8)$$

Outra equação importante, presente em Moukalled *et al.*, (2016) é a equação de transporte de massa, Equação (22), que é usada para modelar o transporte de espécies químicas em fluidos. Essa equação considera a fração mássica da espécie química  $Y$ , sua velocidade molar  $V$ , o coeficiente de difusão  $D$  e a taxa de geração ou consumo da espécie  $R$ .

$$\partial \frac{\rho Y_i}{\partial t} + \nabla \cdot (\rho Y_i V_i) = \nabla \cdot (D \nabla Y_i) + R_i \quad (9)$$

A equação de energia é outra equação fundamental da CFD, que descreve a transferência de energia térmica entre o fluido e o ambiente. Essa equação leva em consideração o calor específico a pressão constante  $C_p$ , a temperatura do fluido  $T$ , a condutividade térmica  $k$  e a taxa de calor gerado no fluido  $Q$  (MOUKALLED *et al.*, 2016).

$$\rho C_p \left( \frac{\partial T}{\partial t} + v \cdot \nabla T \right) = \nabla \cdot (k \cdot \nabla T) + Q \quad (10)$$

Ademais, há várias vantagens distintas do uso de CFD em comparação com abordagens baseadas em experimentos para o projeto de sistemas de fluidos. Versteeg e Malalasekere (1995) apontam como alguma delas a redução significativa dos custos e do

tempo de execução de novos projetos, a capacidade de estudar sistemas que são difíceis ou impossíveis de serem controlados em experimentos, a possibilidade de estudar sistemas em condições perigosas, dentro e fora de seus limites normais de desempenho, além de proporcionar um nível quase ilimitado de detalhes nos resultados obtidos.

## 2.3 ANÁLISE COMPUTACIONAL

A análise computacional é uma técnica amplamente utilizada na simulação de sistemas com fluxo de fluidos, transferência de calor e fenômenos associados, como reações químicas (MOUKALLED *et al.*, 2016). Essa técnica é poderosa e pode ser aplicada em uma ampla gama de áreas, incluindo aeroespacial, automotiva, biotecnologia, processamento de alimentos, energia e meio ambiente (VERSTEEG e MALALASEKERA, 2007).

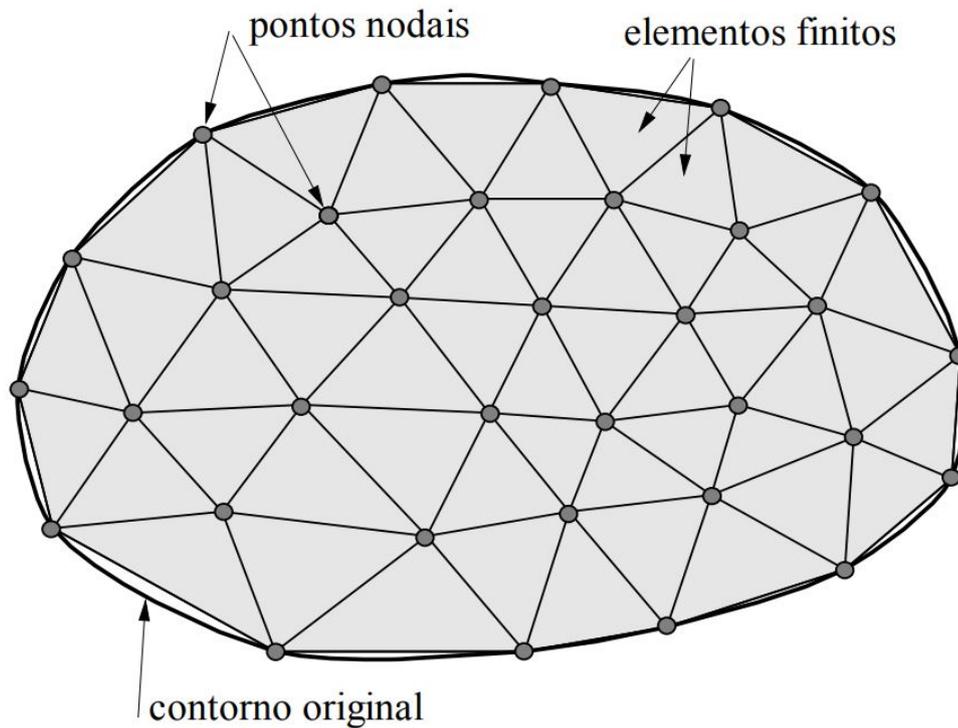
### 2.3.1 Geração de Malha

De acordo com Versteeg e Malalasekera (2007), a malha computacional é um dos principais fatores que afetam a precisão da solução obtida, pois ela influencia diretamente na qualidade da aproximação numérica. Falam também que em análise por elementos finitos, uma malha de elementos finitos, Figura 7, é uma discretização do domínio de um problema em elementos menores para facilitar a análise numérica. Os pontos nodais são os vértices ou pontos de interseção entre os elementos finitos. Eles são usados para definir as variáveis desconhecidas ou parâmetros do problema em cada localização da malha. Os elementos finitos são as formas geométricas (geralmente polígonos ou poliedros) que compõem a malha. Eles são escolhidos de acordo com a natureza do problema e podem variar em forma e tamanho. Dentro de cada elemento, equações matemáticas são usadas para descrever o comportamento do problema, e essas equações são resolvidas numericamente. O contorno original refere-se à fronteira do domínio do problema. É a parte da malha onde são aplicadas condições de contorno, como restrições de deslocamento, forças aplicadas e outros limites físicos do problema. O contorno original desempenha um papel crucial na solução do problema, pois influencia diretamente o comportamento da malha e as soluções numéricas obtidas.

O método de volume finito abordado é, segundo Versteeg e Malalasekera (2007), uma abordagem numérica que se beneficia diretamente da geração de malha. No método de volume finito, o domínio é dividido em células ou volumes finitos, e as equações de conservação são aplicadas em cada uma dessas células. São amplamente utilizados na simulação numérica de uma variedade de fenômenos físicos, incluindo problemas de dinâmica dos fluidos, transferência de calor, transferência de massa e outros processos de transporte. Eles oferecem a vantagem de representar fisicamente a conservação das quantidades físicas.

Thompson *et al.* (2002) afirmam que a qualidade da malha é crucial para garantir

Figura 7 – Malha de Elementos Finitos



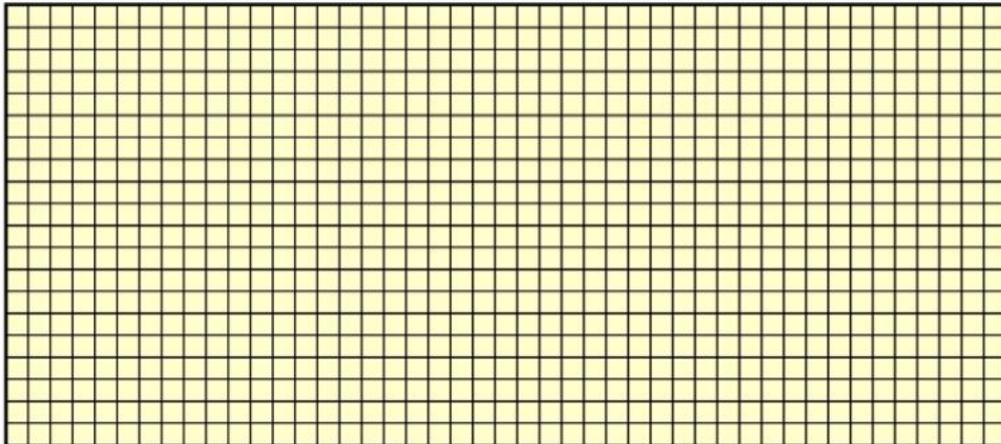
Fonte: Souza (2003).

que os resultados da simulação sejam confiáveis e precisos. Uma malha com baixa qualidade pode levar a erros na solução numérica e pode limitar a resolução do problema. Por outro lado, uma malha de alta qualidade é essencial para obter resultados precisos e confiáveis.

Também é apontado por Versteeg e Malalasekera (2007) que a escolha do tipo de malha a ser utilizado depende da complexidade da geometria do problema e das condições de contorno. A malha estruturada é indicada para problemas com geometria regular e simétrica, enquanto a malha não estruturada é mais adequada para problemas com geometria irregular. Além disso, o autor ressalta que é importante levar em consideração o número de elementos da malha, uma vez que malhas muito refinadas podem aumentar significativamente o tempo de processamento e consumo de recursos computacionais.

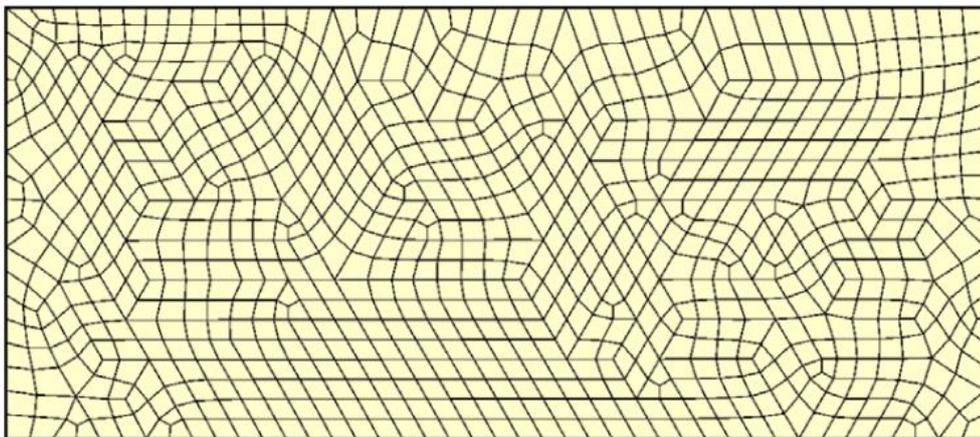
Existem diferentes tipos de malha, cada um adequado para diferentes tipos de geometria e problemas (VERSTEEG e MALALASEKERA, 2007). A malha estruturada, Figura 8, é composta por elementos retangulares, organizados em uma grade regular e é a mais adequada para problemas em que a geometria é regular e simétrica (VERSTEEG e MALALASEKERA, 2007). A malha não estruturada, Figura 9, por outro lado, é mais flexível e pode ser gerada para qualquer tipo de geometria, mas pode ser menos precisa do que a malha estruturada (THOMPSON *et al.*, 2002).

Figura 8 – Malha Estruturada



Fonte: Hurtado (2005).

Figura 9 – Malha Não Estruturada



Fonte: Hurtado (2005).

### 2.3.2 Modelos de CFD

Além disso, existem diferentes tipos de modelos usados no CFD. O modelo de turbulência é usado para simular a turbulência do fluido, que é um fenômeno complexo que ocorre em muitos sistemas (MOUKALLED *et al.*, 2016). O modelo de transporte de massa é usado para simular o transporte de espécies químicas em fluidos, enquanto o modelo de transferência de calor é usado para simular a transferência de calor entre o fluido e o ambiente (VERSTEEG e MALALASEKERA, 2007). O modelo de combustão é usado para simular a combustão de espécies químicas em fluidos (THOMPSON *et al.*, 2002).

Além dos modelos mencionados, existem outras técnicas e métodos de análise computacional que podem ser usados em diferentes tipos de problemas. Por exemplo, a

CFD é frequentemente usada para resolver problemas que envolvem o fluxo de fluidos, como a análise de escoamento em tubulações ou o projeto de aerodinâmica de aeronaves (VERSTEEG e MALALASEKERA, 2007).

De acordo com Versteeg e Malalasekera (2007), existem diferentes tipos de modelos de turbulência disponíveis, cada um com suas próprias características. Entre os modelos mais comuns estão o modelo  $k - \epsilon$ , o modelo  $k - \omega$ , o modelo *Reynolds Stress*, o modelo *Large Eddy Simulation* (LES) e o modelo *Direct Numerical Simulation* (DNS). O modelo  $k - \epsilon$  é amplamente utilizado na indústria devido à sua simplicidade de implementação e capacidade de fornecer resultados razoáveis em muitas situações (MOUKALLED *et al.*, 2016). No entanto, ele pode não ser tão preciso em escoamentos complexos, como escoamentos separados e recirculação. O modelo  $k - \omega$  é menos suscetível a problemas de estabilidade numérica e pode ser mais preciso em escoamentos complexos do que o modelo  $k - \epsilon$  (VERSTEEG e MALALASEKERA, 2007). No entanto, ele é mais difícil de implementar e requer mais recursos computacionais.

Já o modelo *Reynolds Stress* é um modelo de turbulência mais avançado que leva em consideração as flutuações do tensor de Reynolds, tornando-o mais preciso em escoamentos complexos (MOUKALLED *et al.*, 2016). No entanto, ele é mais complexo e computacionalmente intensivo do que o modelo  $k - \epsilon$ . Os modelos LES e DNS são ainda mais avançados e são usados em aplicações em que é necessário prever a turbulência em escalas menores. O modelo LES resolve diretamente as escalas maiores da turbulência e modela as escalas menores, enquanto o modelo DNS resolve todas as escalas da turbulência, tornando-o extremamente preciso, mas também extremamente caro computacionalmente (VERSTEEG e MALALASEKERA, 2007).

Em suma, a escolha do modelo de turbulência adequado depende do tipo de escoamento que está sendo estudado e dos recursos computacionais disponíveis. Cada modelo tem seus próprios pontos fortes e fracos e deve ser escolhido com base nas necessidades da simulação (THOMPSON *et al.*, 2002). Com o uso adequado desses modelos, pode-se prever com precisão a turbulência em uma ampla variedade de aplicações industriais e científicas.

### 2.3.3 Visão Geral do Software

O Ansys Fluent é um software amplamente utilizado para simulação de dinâmica dos fluidos computacional (CFD). Quando se deseja resolver um problema utilizando esse programa, é necessário seguir alguns passos básicos. Primeiramente, é preciso definir os objetivos do modelo a ser criado. Em seguida, deve-se criar a geometria do modelo e a malha que será utilizada para a simulação. Na terceira etapa, é necessário configurar o solver e os modelos físicos que serão utilizados. O próximo passo consiste em executar a simulação e monitorar a solução obtida. Após isso, os resultados podem ser examinados e salvos. Caso seja necessário, é possível realizar ajustes nos parâmetros do modelo físico ou

numérico. Para a segunda etapa do processo de solução, é possível utilizar o DesignModeler e Ansys Meshing dentro do próprio Ansys Workbench, ou um sistema CAD separado para a criação da geometria e malha. Ao trabalhar com geometrias em 3D, também é possível utilizar o modo de malha do próprio Ansys Fluent (ANSYS, 2023).

### 3 METODOLOGIA

Esse trabalho trata-se de uma pesquisa classificada como aplicada, com a finalidade de reduzir o tempo de simulação dinâmica dos fluidos com base no trabalho de Su *et al.* (2022). O objetivo é de desenvolver uma rede neural que seja capaz de prever o coeficiente de variação do hidrogênio. Foi realizada uma abordagem quantitativa através de métodos numéricos e computacionais e procedimento de fontes livros, artigos e outros textos de caráter científico já publicados. O método científico utilizado será o dedutivo. As próximas seções abordarão as atividades e os procedimentos para a geração, a coleta e a organização dos dados da pesquisa basenando-se nas características da simulação de Su *et al.* (2022).

#### 3.1 PROCEDIMENTOS DE COLETA DE DADOS

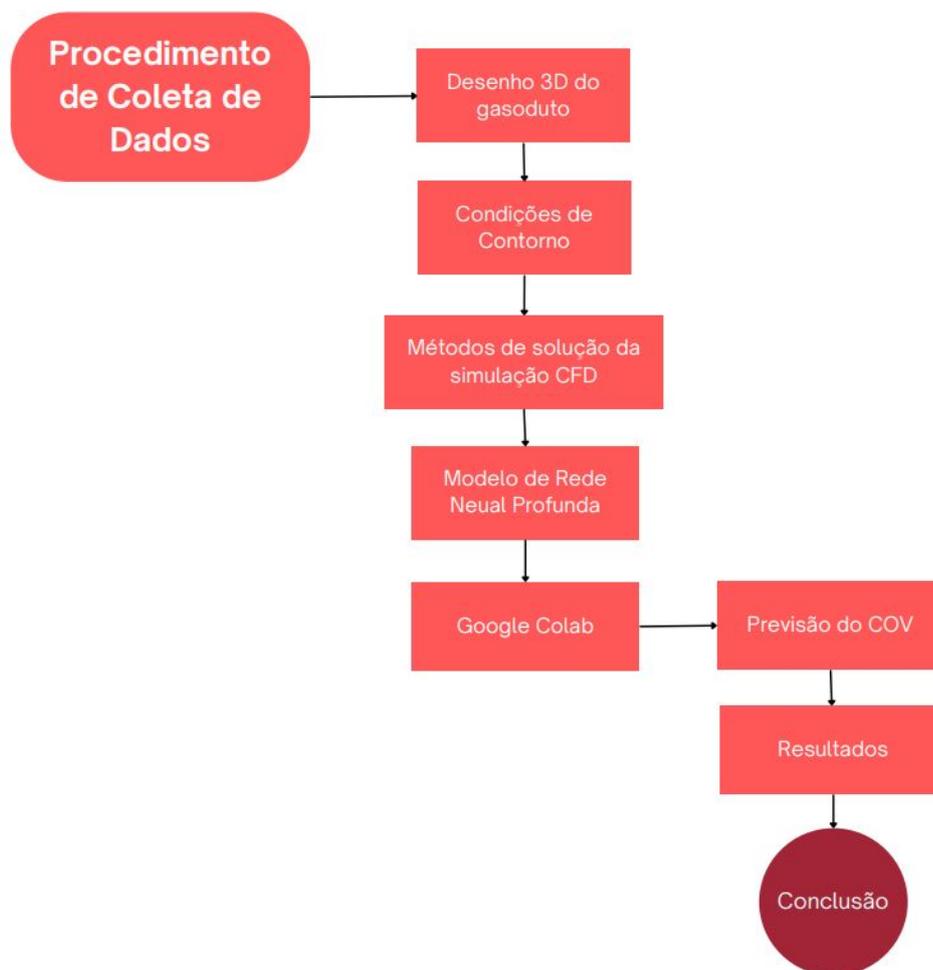
A metodologia adotada pode ser vista na Figura 10. Foi utilizado o software comercial Ansys (3D) para a simulação do fenômeno da injeção de hidrogênio em gasodutos de aços com o transporte de gás natural. O pré-processamento consistirá no desenho tridimensional de trechos destes gasodutos utilizando a ferramenta de projeto e desenho da Ansys, que se denomina *SpaceClaim*, seguindo-se da criação da malha computacional através da ferramenta *Ansys Mesh* e a configuração da simulação no software *Ansys Fluent*, definindo-se os modelos numéricos, matemática, condições iniciais, operacionais e de contorno. No processamento, o *Ansys Fluent* irá calcular a solução do problema analisado e no pós-processamento, serão avaliadas características da composição da mistura.

Para realizar a simulação, é preciso considerar diversas variáveis e condições de contorno. As variáveis que devem ser levadas em conta incluem a composição química da mistura de gás transportada pelo gasoduto, a temperatura da mistura de gás, a pressão interna do gasoduto, a velocidade do fluxo da mistura de gás, as propriedades mecânicas do material do gasoduto..

Já as condições de contorno para a simulação envolvem as condições de entrada do gasoduto, as condições de saída do gasoduto, as condições nas paredes internas do gasoduto, as condições de simetria, as condições de contorno de transferência de massa e as condições de contorno de movimento de fluido.

O intuito é criar externamente um acelerador por utilização de *machine learning* para encontrar as características do resultado da composição da mistura de maneira coerente e de forma mais rápida. Mais especificamente, será de prever o coeficiente de variação ou COV que reflete a variabilidade ou dispersão dos dados em relação à sua média, onde no contexto do trabalho seria quantificar o grau de uniformidade da mistura de hidrogênio e metano em um sistema de tubulação de gás natural. Uma baixa dispersão indica que a mistura é uniforme, enquanto uma alta dispersão sugere falta de uniformidade na mistura. Isso é fundamental para garantir que os gases estejam adequadamente misturados para evitar problemas de segurança, eficiência e desempenho do sistema.

Figura 10 – Metodologia de Pesquisa



Fonte: Autor.

## 3.2 CARACTERÍSTICAS DO MODELO

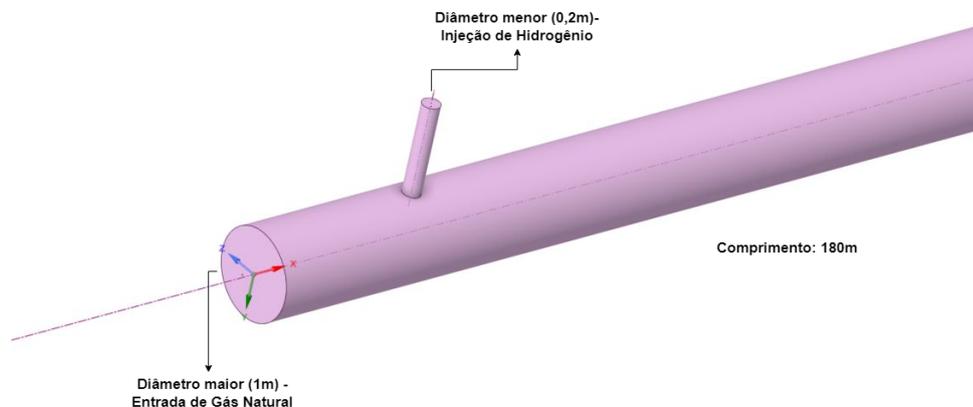
### 3.2.1 Geometria de Simulação

A injeção de hidrogênio em um duto de gás natural será realizada em um duto conforme as características do trabalho de Su *et al.* (2022). Essa injeção será realizada por meio de uma ramificação em forma de T, uma configuração visualmente representada como vista na Figura 11. O duto principal, por onde normalmente flui o gás natural, possui um diâmetro 1m e um comprimento total de 180m.

A ramificação em forma de T, por outro lado, é onde ocorre a injeção controlada de hidrogênio. Esta ramificação tem um diâmetro 0,2m e foi estrategicamente posicionada na parte inferior do duto de gás natural. O posicionamento na parte inferior é fundamental para que a injeção ocorra contra o sentido da ação da gravidade. As características do tubo são vistas na Tabela 1.

A lógica por trás dessa disposição é facilitar a mistura eficiente entre o hidrogênio e o gás natural. Ao permitir que o hidrogênio seja introduzido na parte inferior, onde

Figura 11 – Geometria da Simulação



Fonte: Autor.

Tabela 1 – Características do Duto

Comprimento	180 m
Diâmetro maior	1 m
Diâmetro ramificação	0,2 m

Fonte: Adaptado de Su *et al.* (2022).

as direções do fluxo são distintas, espera-se que a mistura entre os dois gases seja mais eficaz. Isso é crucial, uma vez que uma mistura adequada é essencial para garantir que o hidrogênio seja distribuído uniformemente no fluxo de gás natural.

### 3.2.2 Matemática da Simulação

A fim de simplificar, o gás natural da simulação é representado como metano, que é o principal componente de um gás natural, e com uma fração volumétrica geralmente acima de 90% que seria uma outra simplificação conveniente que reflete a composição típica do gás natural encontrado na maioria das aplicações práticas. O gás natural é uma mistura complexa de hidrocarbonetos, mas o metano é o componente predominante e, em muitos casos, representa a maior parte do gás natural extraído. Os níveis de metano geralmente excedem 90% em termos de fração volumétrica em amostras reais de gás natural.

O modelo matemático tridimensional da simulação no *Ansys* visto no trabalho de Su *et al.* inclui várias equações para descrever o processo de mistura. Entre elas há a equação da continuidade, que descreve a conservação da massa no sistema, levando em

consideração a densidade da mistura de gases, velocidade do fluxo e tempo.:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \quad (11)$$

Onde  $\rho$  é a densidade da mistura gasosa,  $t$  o tempo e  $u_i$  o componente da velocidade ao longo das direções de  $i$  ( $x$ ,  $y$  ou  $z$ ).

A equação de momento que descreve a quantidade de movimento e leva em conta a pressão, viscosidade efetiva e força gravitacional devido à diferença de densidade entre o hidrogênio e o metano (gás natural), é tida por:

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_j} + \frac{\partial}{\partial x_j} \left( \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) + \rho f_i \quad (12)$$

Nela  $p$  é pressão,  $\mu$  viscosidade dinâmica efetiva,  $f_i$  é a força do corpo gravitacional que atua sobre o fluxo de grás, representao efeito de empuxo do hidrogênio e do metano devido à diferença de densidade.

A mistura de hidrogênio e metano é um processo de fluxo de fluido multicomponente. A equação de transporte de componentes descreve a transferência de massa entre o hidrogênio e o metano e leva em consideração a difusão:

$$\frac{\partial(\rho c_s)}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j c_s) = \frac{\partial}{\partial x_j} \left( D_s \rho \frac{\partial c_s}{\partial x_j} \right) \quad (13)$$

De diferente tem-se  $c_s$  que é a concentração volumétrica do componente gasoso  $s$ , enquanto  $D$  é o coeficiente de difusão do componente do gás no ar.

Outro ponto abordado é a Equação de Energia que descreve a variação de temperatura durante o processo de mistura, considerando a condutividade térmica, calor específico a volume constante e termos de fonte de calor:

$$\frac{\partial(\rho c_p T)}{\partial t} + \frac{\partial(\rho c_p u_j T)}{\partial x_j} = \frac{\partial}{\partial x_j} \left( \lambda \frac{\partial T}{\partial x_j} \right) + S \quad (14)$$

Onde  $T$  é a temperatura do gás,  $\lambda$  é o coeficiente de condutividade térmica,  $c_v$  é o calor específico a volume constante e  $S$  é o termo fonte, uma contribuição adicional de energia ou calor que é introduzida ou gerada dentro do sistema.

O Modelo de Turbulência escolhido pelos autores para simulação foi o  $k - \epsilon$  e foi o mesmo utilizado no trabalho. É uma escolha apropriada para abordar o problema devido a várias vantagens e sua capacidade de capturar com precisão os fenômenos turbulentos essenciais nesse tipo de escoamento. Primeiramente, o Modelo  $k - \epsilon$  é particularmente eficaz em situações envolvendo mistura de fluidos com diferentes densidades, como o hidrogênio e o gás natural, pois considera as variações nas taxas de dissipação de energia cinética turbulenta ( $k$ ) e a taxa de dissipação de energia dissipativa ( $\epsilon$ ) em diferentes regiões do escoamento. Essas variações são críticas para determinar como a turbulência se desenvolve e afeta a mistura dos gases, especialmente quando se lida com densidades contrastantes.

Dessa forma, o Modelo de Turbulência  $k-\epsilon$  oferece uma abordagem robusta e confiável para investigar a dinâmica de mistura entre hidrogênio e gás natural, permitindo a compreensão detalhada dos efeitos turbulentos nesse processo e auxiliando na otimização de sistemas de transporte de gases. Esse modelo é expresso pelas seguintes equações:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_i k)}{\partial x_i} = \frac{\partial}{\partial x_j} \left[ \left( \mu + \frac{\mu_t}{\sigma_k} \frac{\partial k}{\partial x_j} \right) \right] + G_k + G_b - \rho \epsilon - Y_M + S_k \quad (15)$$

$$\frac{\partial(\rho \epsilon)}{\partial t} + \frac{\partial(\rho u_i \epsilon)}{\partial x_i} = \frac{\partial}{\partial x_j} \left[ \left( \mu + \frac{\mu_t}{\sigma_\epsilon} \frac{\partial \epsilon}{\partial x_j} \right) \right] + C_{1\epsilon} \frac{\epsilon}{k} (G_K + C_{3\epsilon} G_b) - C_{2\epsilon} \rho \frac{\epsilon^2}{k} + S_\epsilon \quad (16)$$

Onde  $G_k$  é a geração de energia cinética turbulenta devido aos gradientes de velocidade,  $G_b$  é a geração de energia cinética turbulenta devido à flutuabilidade,  $Y_M$  é a contribuição da dilatação flutuante na turbulência compressível para taxa global de dissipação, enquanto os  $C$ 's são constantes.

### 3.2.3 Geração de Malha

Neste trabalho, foi utilizado o software *Ansys* para criar a malha computacional através da ferramenta *Mesh* do *Workbench*. A malha contém elementos triangulares que são conhecidos por sua flexibilidade e capacidade de representar superfícies complexas de maneira eficaz, o que é especialmente relevante em geometrias de tubulações que podem ter curvas e irregularidades. Além disso, a geração de malhas triangulares é frequentemente mais eficiente em termos computacionais do que outras formas de elementos, como tetraedros ou hexaedros, o que pode reduzir significativamente os requisitos de recursos computacionais, tornando as simulações mais acessíveis em termos de tempo. A malha contém 273177 elementos e 60616 nós nas etapas iniciais da tubulação. Essa quantidade de elementos e nós não condizem com o número que foi utilizado na simulação de Su *et al.* (2022). O motivo é que além do número da qualidade da malha do artigo ser muito elevado o que demoraria ainda mais a execução da simulação, o estudo de convergência de malha para atingir os mesmo valores e qualidade também demoraria. Portanto, partiu-se de valor de qualidade inferior afim de já obter alguns resultados iniciais para o treinamento da rede neural.

### 3.2.4 Método de Solução da Simulação CFD

Algumas configurações do modelo da simulação CFD do trabalho de Su *et al.* (2022) tiveram que ser definidas no *ANSYS Fluent* e serão explicadas a seguir.

*Turbulence Model* - Standard  $k-\epsilon$  com efeitos completos de flutuação. O modelo de turbulência escolhido é o *Standard k-ε*, que é uma abordagem comum para modelar a turbulência em escoamentos. A menção aos "efeitos completos de flutuação" indica que o

modelo considera os efeitos da flutuação da densidade na turbulência, o que é relevante quando se lida com gases com diferentes densidades, como hidrogênio e gás natural.

Tratamento de Parede Próxima (*Near-wall treatment*) - Funções de Parede Padrão: Isso se refere à abordagem utilizada para modelar a camada limite próxima à parede da tubulação. As "Funções de Parede Padrão" são métodos que simplificam o cálculo das propriedades de turbulência perto da parede, tornando a simulação mais eficiente.

Acoplamento Pressão-Velocidade (*Pressure-Velocity Coupling*) - *SIMPLE*: O *SIMPLE* (*Semi-Implicit Method for Pressure-Linked Equations*) é um método numérico amplamente utilizado para acoplar as equações de pressão e velocidade em simulações de escoamento. Isso garante que a pressão e a velocidade sejam calculadas de forma consistente e iterativa.

Discretização Espacial do Gradiente (*Spatial Discretization of Gradient*) - Baseada em Células de Quadrados Mínimos: Isso se refere à abordagem usada para calcular gradientes espaciais nas equações de transporte, como as equações de turbulência. A abordagem de "células de quadrados mínimos" é uma técnica numérica que ajuda a suavizar a discretização e reduzir erros numéricos.

Discretização Espacial da Pressão (*Spatial Discretization of Pressure*) - Segunda Ordem: Isso indica que a discretização espacial da pressão é tratada com um esquema de segunda ordem de precisão, o que aumenta a precisão numérica da simulação.

Discretização Espacial do Momento (*Spatial Discretization of Momentum*) - Segunda Ordem *upwind*: A discretização espacial do momento, que é uma das equações fundamentais da dinâmica de fluidos, é tratada com um esquema de segunda ordem com *upwind* o que significa que ele considera principalmente as informações dos pontos de grade adjacentes ao calcular os gradientes das propriedades do fluido.

Discretização Espacial da Energia (*Spatial Discretization of Energy*) - Segunda Ordem *upwind*: A energia é uma quantidade importante em simulações de escoamento e é tratada com um esquema de segunda ordem com *upwind* para garantir cálculos precisos.

Discretização Espacial da Taxa de Dissipação Turbulenta (*Spatial Discretization of Turbulent Dissipation Rate*) - Primeira Ordem *upwind*: A taxa de dissipação turbulenta é outra propriedade crucial em modelos de turbulência. Ela é tratada com um esquema de primeira ordem "*upwind*," que enfatiza informações dos pontos de grade próximos.

Discretização Espacial da Cinética Turbulenta (*Spatial Discretization of Turbulent Kinetic Energy*) - Primeira Ordem *upwind*: A cinética turbulenta é uma medida importante da turbulência em um escoamento e também é tratada com um esquema de primeira ordem "*upwind*" para cálculos eficientes.

Essas configurações são escolhidas para governar como as equações fundamentais que descrevem o escoamento, a turbulência e a transferência de energia são discretizadas e resolvidas numericamente no software Fluent, garantindo precisão e eficiência na simulação da injeção de hidrogênio na tubulação.

Tabela 2 – Configurações do Modelo

Modelo de Turbulência	Standard k- $\epsilon$
Tratamento Próximo a Parede	Funções padrão de parede
Acoplamento pressão-velocidade	Simples
Discretização espacial do gradiente	Mínimos quadrados
Discretização espacial da pressão	Segunda Ordem
Discretização espacial do momento	Upwind de segunda ordem
Discretização espacial da energia	Upwind de segunda ordem
Discretização espacial da taxa de dissipação turbulenta	Upwind de primeira ordem
Discretização espacial da energia cinética turbulenta	Upwind de primeira ordem

Fonte: Adaptado de Su *et al.* (2022).

### 3.3 MODELO DE REDE NEURAL PROFUNDA

Embora a simulação de Dinâmica dos Fluidos Computacional (CFD) tenha sido utilizada para investigar qualitativamente essa uniformidade em diferentes cenários, ela frequentemente demanda tempo considerável e não é adequada para fornecer respostas rápidas sob várias condições de trabalho na engenharia prática. Para superar essa limitação e viabilizar uma previsão ágil da uniformidade da mistura, esta seção apresenta um modelo de Rede Neural Profunda (DNN) baseado em dados obtidos por simulações CFD.

Durante a etapa de propagação direta, a função de ativação ReLU é aplicada nos nós da rede. A escolha da função ReLU foi pela sua capacidade de introduzir não-linearidade na rede. A ReLU é uma função simples que mantém os valores positivos inalterados e "zera" os valores negativos. Essa não-linearidade é essencial porque permite que a DNN aprenda a representar padrões complexos nos dados. Além disso, a ReLU ajuda a mitigar o problema de desaparecimento de gradientes, comum em redes profundas, garantindo que os gradientes transmitidos durante a retropropagação de erros não se tornem muito pequenos. Em resumo, a função de ativação ReLU é uma escolha eficaz para permitir que a DNN aprenda e represente relacionamentos não lineares nos dados, facilitando o aprendizado de características complexas e a construção de modelos mais poderosos.

A escolha do otimizador foi o Adam para treinar a rede neural neste trabalho é fundamentada em sua eficiência e capacidade de adaptação da taxa de aprendizado. O Adam combina as vantagens do Gradiente Descendente Estocástico (SGD) e oferece uma convergência rápida e estável. O Adam é uma escolha padrão em muitos cenários de aprendizado profundo, proporcionando uma abordagem eficaz e de fácil utilização para otimizar redes neurais.

Su *et al.* (2022) comentam que para construir um modelo DNN preciso e eficiente para prever o COV, foram realizados experimentos para determinar o número ideal de camadas ocultas e o número de nós em cada camada. O objetivo era equilibrar a precisão das previsões com a eficiência computacional e evitar o superajuste do modelo. Com isso,

concluíram que um modelo com 11 camadas ocultas e 40 nós em cada camada alcançava poderiam encontrar melhor esse equilíbrio e foi o norte utilizado para o desenvolvimento da rede deste trabalho.

### 3.3.1 Google Colab

Neste estudo, a metodologia de desenvolvimento e treinamento do modelo de Rede Neural Profunda foi conduzida em um ambiente de desenvolvimento proporcionado pelo Google Colab. O Google Colab é uma plataforma em nuvem que oferece recursos gratuitos de GPU e também TPU para a execução de código *Python* em *Jupyter Notebooks*. Isso proporcionou uma infraestrutura poderosa sem a necessidade de configurações locais para a parte do código da rede. Para construir e treinar o modelo, utilizamos diversas bibliotecas e frameworks, dentre elas a *NumPy*, uma biblioteca essencial para computação numérica em Python, permitindo operações matriciais e manipulação eficiente de dados, a *Pandas* que é uma biblioteca de análise de dados que disponibiliza estruturas de dados versáteis e eficazes para trabalhar com conjuntos de dados tabulares. Também foi usada o *TensorFlow* que é um framework de código aberto desenvolvido pela Google, usado para criar e treinar modelos de aprendizado profundo, ele foi o responsável para construir e treinar a DNN. Um muito utilizado a atrelado ao código foi o *Scikit-Learn* utilizada para machine learning, que oferece ferramentas para pré-processamento de dados, treinamento de modelos e avaliação de desempenho. No código seu uso serviu para dividir os dados em conjuntos de treinamento e validação, bem como normalizar os dados de entrada. E por fim também utilizou-se o *Matplotlib*, uma biblioteca de visualização de dados em Python que nos permitiu criar gráficos para acompanhar o processo de treinamento, incluindo a visualização do erro em relação ao número de épocas.

### 3.3.2 Previsão do ??

Com o modelo configurado com 11 camadas ocultas e 40 nós em cada camada oculta, a previsão do Coeficiente de Variação (COV) da concentração de hidrogênio na seção do duto foi realizada. As entradas do modelo DNN incluíram a razão de mistura de hidrogênio (HBR) em 10%, a temperatura do gás (T) em 300K, a relação entre a distância de fluxo e o diâmetro do duto ( $z/D$ ) que foram divididas em 18 seções de múltiplos do diâmetro da tubulação, a velocidade do fluxo de metano ( $v$ ) de 1m/s e a razão de diâmetro entre o duto de ramificação e o duto de gás natural ( $d/D$ ) que é de 0,2. Características essas também descritas no trabalho de Su *et al.* (2022)

O uso do modelo DNN para prever o COV proporciona uma avaliação ágil e precisa da uniformidade da mistura de hidrogênio e gás natural em diversas condições de trabalho na engenharia prática. A função de ativação ReLU, devido à sua natureza não linear, foi considerada crucial para a eficácia do modelo, permitindo que ele aprenda padrões complexos nos dados e forneça previsões confiáveis. Essa abordagem busca aprimorar a

Tabela 3 – Dados de Entrada da Rede Neural

Razão de Mistura de Hidrogênio	10%
Temperatura do Gás	300K
Velocidade Metano	1m/s
Razão Duto de Ramificação e Duto de Gás	0,2

Fonte: Adaptado de Su *et al.* (2022)

segurança e eficiência do transporte de gás natural com hidrogênio, permitindo a rápida adaptação às diferentes situações operacionais.

## 4 RESULTADOS E DISCUSSÕES

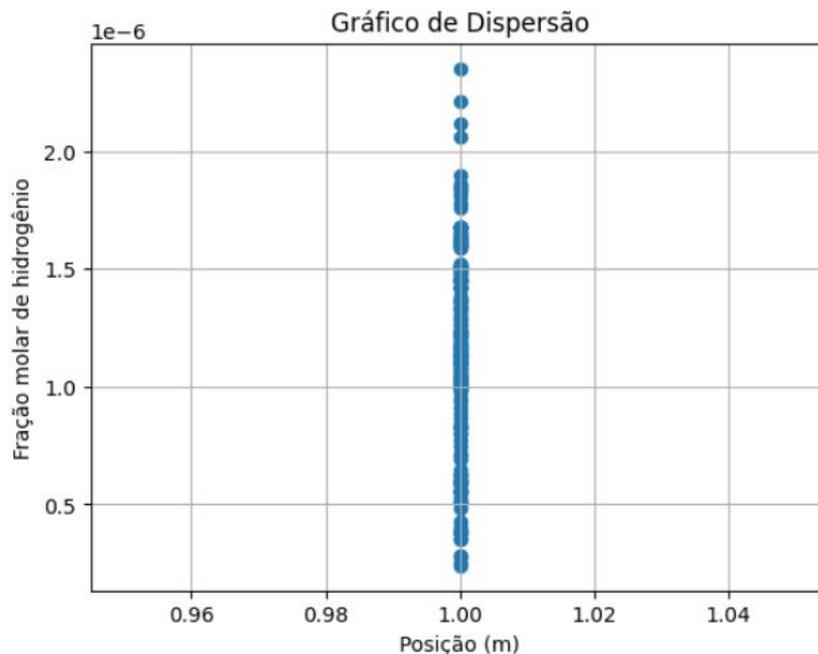
### 4.1 DETERMINAÇÃO DO COV

O valor específico do COV para melhor determinar a qualidade da mistura do hidrogênio com o gás natural não é uma saída bem especificada da simulação. A expressão em que o COV pode ser estimado pode ser vista na Equação (17).

$$COV = \frac{1}{\bar{c}} \sqrt{\frac{\sum_{i=1}^n (c_i - \bar{c})^2}{n - 1}} \quad (17)$$

Onde  $c_i$  é a fração volumétrica da concentração de hidrogênio no ponto de amostragem e  $\bar{c}$  é a fração volumétrica média da concentração de hidrogênio em todos os pontos de amostragem,  $n$  é o número de pontos de amostragem que no caso foram as 18 seções do tubo. Su *et al.* (2022) comentam que quanto maior o valor de  $n$ , mais preciso é o COV. Neste estudo, foi determinada as 18 seções equidistantes proporcionais a um múltiplo do diâmetro assim como no trabalho mencionado. Outro ponto que abordam é que o hidrogênio e o gás natural podem ser considerados completamente misturados e uniformes quando o COV é menor ou igual a 0,05. Com isso, para a determinação do COV da simulação é preciso dos dados da fração volumétrica. No entanto, esse é um outro dado que não é exposto com clareza na simulação tão pouco houve uma descrição detalhada da sua obtenção no artigo em questão, para isso será determinado a partir da fração molar que a simulação dispõe como visto na Figura 12 um exemplo do dado na seção de 1 metro do trecho do tubo.

Figura 12 – Fração Molar na Seção de 1m do Tubo.



Fonte: Autor.

Ela obedece a Equação (18):

$$\%n_i = \frac{n_i}{n_t} \cdot 100 \quad (18)$$

Onde  $n_i$  o número de mols do trecho,  $n_t$  o número de mols total da mistura e  $\%n$  a porcentagem molar no trecho, sendo esse último o dado disponível na simulação. A fração volumétrica que precisa ser encontrada é dada por Equação (19).

$$\%V_i = \frac{V_i}{V_{trecho}} \cdot 100 \quad (19)$$

Sendo  $V_i$  o número do volume do componente,  $V_{trecho}$  o volume total do trecho da mistura e  $\%V$  a porcentagem em volume no trecho.

Para a determinação das frações volumétricas, foram utilizados os trechos iniciais da tubulação onde a mistura era composta em sua maioria ainda pelo gás metano. Então, para o cálculo foi considerado trechos a cada metro da tubulação em seu início onde a presença apenas de metano era mais evidente como visto na Equação (20).

$$n_{CH_4} = \frac{PV_{trecho}}{RT} \quad (20)$$

Para os cálculos dos trechos subsequentes os valores alterados eram o volume total do trecho e acrescentou-se um fator proporcional da diferença da fração molar do metano com a fração molar do hidrogênio do trecho analisado. Com a consideração do número de mols, calculou-se o número de mols de H<sub>2</sub> a partir da fração molar extraída da simulação na Equação (21)

$$n_{H_2} = (\%n_{CH_4} - \%n_{H_2}) \cdot n_{CH_4} \quad (21)$$

Considerando que o H<sub>2</sub> tem 2g/mol, esse número foi utilizado para obter como produto a massa de H<sub>2</sub> presente no trecho (Equação (22)). E considerando H<sub>2</sub> como gás ideal e as condições da simulação de 0.3MPa e 300K estimou-se a densidade com a equação Equação (23).

$$m_{H_2} = 2\text{g/mol} \cdot n_{H_2} \quad (22)$$

$$\rho_{H_2} = \frac{n}{V} = \frac{P}{RT} \quad (23)$$

O número do volume do componente foi então definido pela Equação (24).

$$V_i = \frac{m_{H_2}}{\rho_{H_2}} \quad (24)$$

Enfim, com o volume do trecho pode-se encontrar cada fração volumétrica de cada trecho da tubulação pela Equação (25).

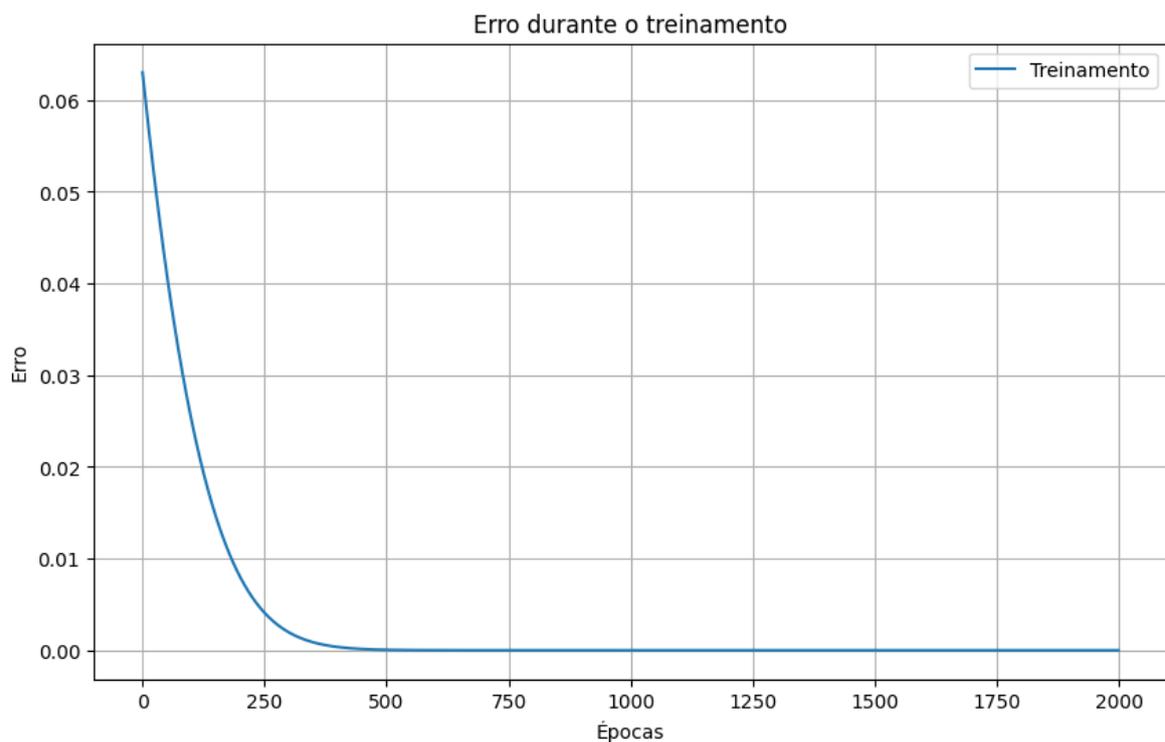
$$c_i = \frac{V_i}{V_{trecho}} \quad (25)$$

Com isso foi calculado a fração volumétrica de cada trecho e encontrou-se também a fração volumétrica média e a estimativa do COV da Equação (17) pode ser feita e os resultados foram inseridos na camada de saída da rede para ser feito o treinamento.

## 4.2 TREINAMENTO DA REDE

O código desenvolvido pode ser encontrado neste [link](#). Para realizar essa tarefa de previsão, utilizamos um conjunto de dados de treinamento que inclui parâmetros como HBR (Taxa de Hidrogênio), velocidade, temperatura, diâmetro da tubulação e razão de distância de fluxo. Além disso, incorporou-se os resultados do COV correspondentes a esses parâmetros. O modelo DNN foi projetado com 11 camadas ocultas, cada uma contendo 40 nós com a função de ativação ReLU. A escolha da profundidade e largura da rede foi baseada em experimentações de Su *et al.* (2022). Os primeiros testes contou com camada de saída com uma única unidade de ativação linear, pois estamos prevendo um valor contínuo. Para otimizar nosso modelo DNN, utilizou-se Adam e a métrica de erro médio quadrado (MSE) como função de perda. O treinamento do modelo foi executado inicialmente com com 100 épocas depois aumentado até 2000 épocas, que foi ajustado iterativamente para minimizar o erro até obter um dos resultados presentes na Figura 13.

Figura 13 – Treinamento da Rede



Fonte: Autor

### 4.2.1 Verificação de Resultados

A fim de avaliar a eficácia do modelo da rede desenvolvido para prever o Coeficiente de Variação (COV) na mistura de hidrogênio e gás natural, foi feita uma fase de verificação. Essa etapa que configura-se como essencial para garantir a confiabilidade e a precisão das previsões do modelo.

Para realizar essa verificação, foram realizadas novas simulações que envolveram a alteração intencional de alguns dados de entrada. Essas alterações visavam representar diferentes cenários operacionais e condições de fluxo que poderiam ocorrer na prática. Entre os parâmetros ajustados estavam a Taxa de Recirculação de Hidrogênio (HBR), a velocidade do fluxo, a temperatura do gás, a relação entre o diâmetro da tubulação e o diâmetro do duto principal ( $d/D$ ), bem como a relação entre a distância de fluxo e o diâmetro do duto ( $z/D$ ).

A princípio, o parâmetro alterado para checar o resultado da predição da rede foi a velocidade do fluxo do gás. A decisão foi puramente por em cenários reais esse seria um parâmetro mais fácil de ser alterado do que mudar o diâmetro do tubo por exemplo.

Essa abordagem de validação cruzada, com simulações baseadas em diferentes cenários operacionais, permitiu verificar a robustez e a generalização do modelo. Confirmar que o modelo pode lidar com uma variedade de entradas é fundamental para sua aplicabilidade em situações do mundo real.

Os resultados dessas simulações de verificação e validação são apresentados e discutidos na seção de resultados deste trabalho, contribuindo para uma compreensão mais abrangente do desempenho da rede neural na previsão do COV em condições diversas.

### 4.2.2 Análise dos Resultados

Após utilizar os dados dos trechos da simulação da fração molar de hidrogênio, estimar a fração volumétrica e calcular o cov para usar para dados de treinamento da rede, foi então utilizada a rede para buscar prever os valores de COV com parâmetros de entrada fora do escopo do treinamento. Nessas outras simulações em que alterou-se o valor da entrada da velocidade, os valores previstos pelo modelo não correspondiam inteiramente aos valores COV que foram recalculados pelo método anteriormente descrito. Essa discrepância levantou questões importantes e merece uma análise aprofundada para compreender as possíveis razões por trás dessas inconsistências.

Em um dos casos das simulações feitas, alterando o dado de entrada da velocidade para 10m/s, obteve-se um valor ao usar a rede treinada com os primeiros dados de  $1,52 \times 10^{-1}$ , como visto na Tabela 4.

Usando esse exemplo dentre outros das primeiras execuções da rede, esse resultado se mostra insatisfatório por dois motivos. O primeiro é que para o treinamento da rede a velocidade de 1m/s estava inclusa no escopo do treinamento e esse valor condizia

com o valor que era calculado pelo método descrito anteriormente e pode ser visto na Tabela 4. Outro fator que poderia indicar o erro caso fosse um valor que estivesse fora de conhecimento, é que com o aumento da temperatura a rede estava prevendo valores menores de COV, e como pode ser visto em Su *et al.* (2022) a tendência é que o COV aumente e o gás se torne mais disperso.

Conforme os resultados retirados da simulação se mostravam diferentes dos presentes na rede, o procedimento adotado foi de que cada vez mais esses novos dados de entrada e o COV calculado da nova simulação fossem utilizados para novos treinamentos da rede neural.

Apesar disso, por mais que o valor passasse a se aproximar do que era previsto, ou seja quanto maior fosse a velocidade maior estivesse sendo o COV, no cálculo essa aproximação ainda estava longe do resultado e provavelmente muito mais treinamento seria necessário. Isso pode ser visto na imagem do erro de treinamento da figura Figura 13, onde há um grande intervalo de amortização. Como os valores não estavam se aproximando uma das decisões tomadas foi de aumentar o número de épocas do treinamento da rede, mas como é visto na imagem o quanto boa parte do período das épocas já estavam bem amortizadas e a rede não estava predizendo bem, seria um indício de que não havia dados o suficiente para o treinamento.

Voltando no exemplo da mudança do dado de entrada para 10m/s, com novos treinamentos o valor do COV tendeu a ser maior que o primeiro, ou seja aumentou como o cálculo indicava estudo de Su *et al.* (2022) indicava que aumentaria, mas ainda estava distante do valor real, como visto na ??.

Tabela 4 – Resultados da Predição

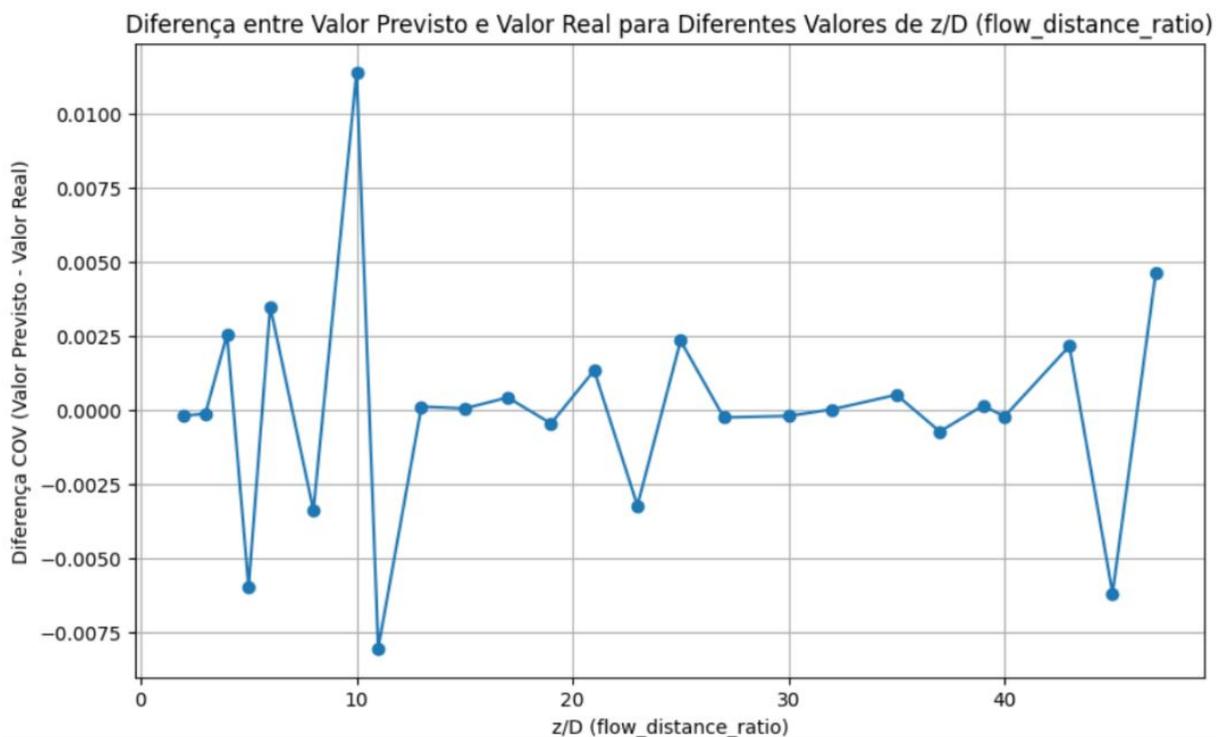
COV para 10m/s	$1,52 \times 10^{-1}$
COV para 1m/s	$2,51 \times 10^{-1}$
COV para 10m/s com maior treinamento	$5.31 \times 10^{-1}$

Fonte: Autor.

Uma explicação possível para as diferenças de resultados, vista melhor na Figura 14, pode ser a qualidade e a representatividade dos dados de treinamento que, embora tenha sido feito o possível para coletar dados precisos e abrangentes, a restrição da coleta dos dados nos primeiros trechos da simulação devido a grande demanda computacional deve ter se mostrado insuficiente para a representatividade completa de todas as condições operacionais possíveis. O modelo DNN aprende com os dados de treinamento disponíveis e pode não ser capaz de generalizar completamente para os cenários não observados a frente.

A escolha dos hiperparâmetros do modelo, como o número de camadas ocultas e o número de nós em cada camada, também desempenha um papel crítico no desempenho da rede. Embora tenhamos seguido as recomendações do estudo original, é possível que

Figura 14 – Diferença do valor real pelo valor previsto na rede



Fonte: Autor.

ajustes adicionais fossem necessários para otimizar ainda mais a precisão do modelo para as condições de simulação que tinha em disposição.

Além disso, as características dos dados de entrada, como a relação entre a distância de fluxo e o diâmetro da tubulação ( $z/D$ ) e a relação entre o diâmetro do ramal da tubulação e o diâmetro da tubulação principal ( $d/D$ ), podem ser particularmente sensíveis às variações e desafiar a capacidade de previsão do modelo. Um ponto que poderia ser feito seria aumentar o número de seções para coletar as amostras de frações volumétricas para o treinamento da rede.

Por fim, é importante considerar que o próprio conceito de COV pode ser influenciado por diferentes métodos de cálculo e definições. A escolha de como o COV é calculado pode variar entre diferentes simulações e medições, o que pode resultar em discrepâncias mesmo quando o modelo está operando corretamente. Como não era um valor que pôde ser retirado diretamente da simulação, algumas considerações feitas para chegar ao seu valor podem ter influenciado também.

## 5 CONCLUSÃO

Neste trabalho foi desenvolvida uma rede neural em uma simulação de escoamento de gás natural enriquecido por hidrogênio verde. Para isso, foram selecionadas técnicas de aprendizagem profunda, especificamente uma Rede Neural Profunda, para resolver o problema de previsão do Coeficiente de Variação a mistura de hidrogênio e gás natural em tubulações. A escolha da DNN, com 11 camadas ocultas e 40 nós em cada camada, foi embasada nas melhores práticas identificadas na literatura que já lidaram com a complexidade do problema.

Para a implementação da Rede Neural Profunda (DNN), foi possível cumprir com o objetivo de realizar uma pesquisa detalhada das bibliotecas disponíveis foi feita no ecossistema Python. Duas das principais bibliotecas consideradas e posteriormente selecionadas foram TensorFlow e Keras, devido à sua reputação consolidada na comunidade de aprendizado profundo e às vantagens que oferecem para desenvolvedores e pesquisadores. A escolha dessas bibliotecas foi baseada primeiramente devido a popularidade delas, refletida em sua ampla adoção pela comunidade, indicou sua confiabilidade. Além disso, a documentação abrangente disponível para ambas as bibliotecas desempenhou um papel fundamental, tornando mais acessível o processo de aprendizado e desenvolvimento para a equipe. Por fim, a capacidade dessas bibliotecas de oferecer suporte sólido à implementação de redes neurais profundas foi um fator determinante em nossa escolha, uma vez que a complexidade do problema exigia um ambiente de desenvolvimento flexível e poderoso.

Outro objetivo alcançado foi definir o cenário de simulação que foi utilizado para testar a rede neural desenvolvida conforme o trabalho de Su *et al.* (2022). Os dados de entrada, incluindo HBR, velocidade, temperatura, diâmetro da tubulação, e a relação de distância do fluxo foram os parâmetros de entrada do transporte de hidrogênio e gás natural em tubulações que foram levadas em consideração no desenvolvimento da rede. Esses cenários foram usados como entrada para a rede neural.

A avaliação da eficácia da abordagem proposta foi o objetivo de maior dificuldade. Foram realizadas simulações com e sem a utilização da rede neural para prever o COV. Obtiveram-se valores não muito próximos dos cálculos. A discrepância entre os valores previstos pela rede e os valores reais do COV sugere que há espaço para melhorias na arquitetura da rede ou na coleta de dados de treinamento.

## 6 TRABALHOS FUTUROS

Para aprimorar a eficácia da abordagem proposta, é essencial realizar uma revisão abrangente da arquitetura da rede neural. Isso envolve a otimização dos hiperparâmetros, como a taxa de aprendizado, o número de camadas e neurônios nas camadas ocultas, bem como a escolha adequada de funções de ativação. Essa revisão detalhada permitirá sintonizar a rede neural para que ela possa capturar com mais precisão as nuances das interações entre os parâmetros de entrada e o COV.

Outra ação crucial é aumentar tanto o tamanho quanto a diversidade do conjunto de dados de treinamento. Isso significa coletar dados de simulação de uma ampla variedade de cenários de transporte de hidrogênio e gás natural, abrangendo diferentes faixas de velocidade, temperaturas, diâmetros de tubulação e relações de distância do fluxo. Dados mais representativos ajudarão a rede neural a generalizar melhor, tornando suas previsões mais robustas e aplicáveis a uma variedade de situações reais.

Além disso, é vital explorar a possível inclusão de variáveis adicionais que possam afetar o COV. Isso pode envolver a consideração de parâmetros ambientais, como pressão atmosférica e umidade, ou outros fatores que influenciam o comportamento do gás durante o transporte. Realizar testes mais extensivos e conduzir uma análise estatística mais aprofundada dos resultados obtidos permitirá avaliar a eficácia da abordagem proposta de maneira mais abrangente. Em última análise, a pesquisa e o desenvolvimento contínuos são essenciais para alcançar um modelo de rede neural que seja confiável e eficaz na previsão do COV em cenários de transporte de hidrogênio e gás natural em tubulações, contribuindo assim para a melhoria da eficiência e sustentabilidade desses processos industriais.

## 7 REFERÊNCIAS

AGGARWAL, C. C. **Neural Networks and Deep Learning**. *Springer*, 2018.

ANSYS. **Ansys Fluent User's Guide, Version 2023 R1**. *ANSYS, Inc.*, 2023.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. *Springer*, 2006.

CHOLLET, F. **Deep Learning with Python**. *Manning Publications*, 2018.

DUCHI, J.; HAZAN, E.; SINGER, Y. **Adaptive subgradient methods for online learning and stochastic optimization**. *Journal of Machine Learning Research*, 2011.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 1st edition, O'Reilly Media, 2017.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. *MIT Press*, 2016.

HURTADO F. **Uma formulação de volumes finitos baseada em elementos para a simulação do deslocamento bifásico imiscível em meios porosos**. *ResearchGate*, 2005.

KINGMA, D. P.; BA J. **Adam: A method for stochastic optimization**. *Conference for Learning Representations*, 2015.

KOCHKOV, D. et al. **Machine learning-accelerated computational fluid dynamics**. *Proceedings of the National Academy of Sciences*, [S.l.], v. 118, n. 21, 2021.

LIU, B.; LIU, X.; LU, C.; GODBOLE, A.; MICHAL, G.; TEN, L. **Decompression of hydrogen-natural gas mixtures in high-pressure pipelines: CFD modelling using different equations of state**. *International Journal of Hydrogen Energy*, [S.l.], v. 44, p. 7428-7437, 2019.

MACHADO, T. S.; FONSECA JÚNIOR, N. J. **Redes neurais artificiais aplicadas na previsão do VTEC no Brasil**. *Simpósio Brasileiro de Sensoriamento Remoto*, 2015.

MITCHELL, Tom M. **Does machine learning really work?** *AI Magazine*, v. 18, n. 3, p. 11-11, 1997.

MOUKALLED, F., MANGANI, L.; DARWISH, M. **The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab.** *Springer*, 2016.

NUNES-VILLELA, J.; RAPOZO, F.; DOMINGOS, M. L. C.; QUELHAS, O. L. G. **Energia em tempo de descarbonização: uma revisão com foco em consumidores fotovoltaicos.** *Revista Brasileira de Ciências Ambientais*, [S.l.], n. 45, p. 130-144, 2017. DOI: 10.5327/Z2176-947820170264. Disponível em: [https://www.abesdn.org.br/publicacoes/rbciamb/Ed45/RBCIAMB\\_n45\\_130-144.pdf](https://www.abesdn.org.br/publicacoes/rbciamb/Ed45/RBCIAMB_n45_130-144.pdf). Acesso em: 13 dez. 2022.

RASCHKA, S.; MIRJALILI, V. **Python Machine Learning.** *Packt Publishing*, 2020.

ROBERTS, Daniel A., Sho Yaida, and Boris Hanin. **The principles of deep learning theory.** Cambridge, MA, USA: Cambridge University Press, 2022.

RUDER, S. **An overview of gradient descent optimisation algorithms.** 2016. Disponível em: <https://www.ruder.io/optimizing-gradient-descent/>. Acesso em: 01 abr. 2023.

RUMELHART, D. E., HINTON, G. E.; WILLIAMS, R. J. **Learning representations by back-propagating errors.** *Springer*, 1986.

SOUZA, R. **O Método dos Elementos Finitos Aplicado ao Problema de Condução de Calor.** *ResearchGate*, 2003.

Su, Y.; Li, J.; Guo, W.; Zhao, Y.; Li, J.; Zhao, J.; Wang, Y. **Prediction of Mixing Uniformity of Hydrogen Injection in Natural Gas Pipeline Based on a Deep Learning Model.** *Energies* 2022.

THOMPSON, J. F., Soni, B. K.; Weatherill, N. P. **Handbook of grid generation.** *CRC press*, 2002.

VERSTEEG, H. K.; MALALASEKERA, W. **An introduction to computational fluid dynamics: the finite volume method** *Pearson Education Limited*, 2007.

ZAKARIA, Magdi, A. S. Mabrouka, and Shahenda Sarhan. **Artificial neural network: a brief overview.**, 2014.