



Universidade Federal
de São João del-Rei



UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL REI
PROFMAT - MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL

SIMONE MYLENA DE MELO

REDE NEURAL PERCEPTRON MULTICAMADAS NO
DESENVOLVIMENTO DE UM APLICATIVO PARA
TRIAGEM DE TEA EM ADOLESCENTES

SÃO JOÃO DEL REI
2025

SIMONE MYLENA DE MELO

**REDE NEURAL PERCEPTRON MULTICAMADAS NO
DESENVOLVIMENTO DE UM APLICATIVO PARA
TRIAGEM DE TEA EM ADOLESCENTES**

Dissertação apresentada à Universidade Federal de São João del Rei, Campus Santo Antônio, como parte das exigências do Programa de Pós-Graduação Mestrado Profissional em Matemática em Rede Nacional, para obter o título de Mestre.

Orientador

Jorge Andrés Julca Avila

SÃO JOÃO DEL REI
2025

Ficha catalográfica elaborada pela Divisão de Biblioteca (DIBIB)
e Núcleo de Tecnologia da Informação (NTINF) da UFSJ,
com os dados fornecidos pelo(a) autor(a)

M528r Melo, Simone Mylena de.
Rede Neural Perceptron Multicamadas no
Desenvolvimento de um Aplicativo para Triagem de TEA
em Adolescentes / Simone Mylena de Melo ; orientador
Jorge Andrés Julca Avila. -- São João del-Rei, 2025.
106 p.

Dissertação (Mestrado - Programa de Mestrado
Profissional em Matemática em Rede Nacional -
PROFMAT) -- Universidade Federal de São João del
Rei, 2025.

1. Redes Neurais Artificiais. 2. Transtorno do
Espectro Autista. 3. Aprendizado de Máquina. 4.
Aplicativo. 5. Triagem. I. Avila, Jorge Andrés
Julca, orient. II. Título.

SIMONE MYLENA DE MELO

**REDE NEURAL PERCEPTRON MULTICAMADAS NO
DESENVOLVIMENTO DE UM APLICATIVO PARA
TRIAGEM DE TEA EM ADOLESCENTES**

Dissertação apresentada à Universidade Federal de São João del Rei, Campus Santo Antônio, como parte das exigências do Programa de Pós-Graduação Mestrado Profissional em Matemática em Rede Nacional, para obter o título de Mestre.

APROVADA: 11 de novembro de 2025.

Jorge Andrés Julca Avila
(Orientador)

Juan Carlos Zavaleta Aguilar - UFSJ
(Membro Interno do PROFMAT)

Santos Alberto Enriquez Remigio - UFU
(Membro Externo)

SÃO JOÃO DEL REI
2025

Aos pais e às pessoas com Transtorno do Espectro Autista. Que este trabalho seja uma pequena contribuição para a compreensão, o respeito e a inclusão. Vocês que enfrentam o desconhecido com coragem e amor, saibam que não estão sozinhos.

Agradecimentos

Agradeço, primeiramente, a Deus, por sempre se mostrar presente em minha vida. À Santa Teresinha e à Nossa Senhora Aparecida, por serem minhas intercessoras e amigas no céu. Sei que estavam torcendo por mim.

Agradeço também, de todo coração, ao meu maior presente de Deus: minha família, minha base, minha maior fonte de felicidade, paz e amor. Aos meus pais, José Luiz e Celina, meus maiores exemplos de amor e cuidado, que sempre me ensinaram os verdadeiros valores da vida. Eu não seria nada sem vocês. Ao meu marido, Luan, que, com amor, carinho e paciência, esteve presente em cada detalhe desta jornada. Às minhas irmãs, Carina e Naiane, por sempre me incentivarem a continuar e me mostrarem que sou capaz. Aos meus sobrinhos e afilhados João Henrique, Luísa, Laís, Henrique, Conrado e Alice, por sempre trazerem tanta felicidade e leveza ao longo do processo, muitas vezes, era tudo o que eu precisava. Ao meu avô, Palmério de Melo, que sempre lutou tanto pela educação. Ao concluir este trabalho, realizo um sonho profissional e pessoal, uma conquista que não é apenas minha, mas de toda a minha família.

Tenho infinitos motivos para agradecer. Outro deles é o meu querido orientador, professor Jorge, um educador humano e amigo que, além de ensinar e inspirar, me ajudou e motivou para que este trabalho fosse feito da melhor forma possível. Agradeço também à banca avaliadora e a todos os professores da Universidade Federal de São João del-Rei que fizeram parte da minha trajetória no Mestrado Profissional em Matemática.

Agradeço ainda às minhas amigas de jornada, Cindy e Edilamar. Compartilhar essa caminhada com vocês tornou tudo mais leve e feliz.

Por fim, agradeço aos meus alunos (atuais, ex-alunos e até futuros). É por eles que busco, todos os dias, novos conhecimentos e formas de aprimorar minhas metodologias, com o desejo de oferecer uma educação com qualidade, empatia e carinho.

Concluo este trabalho com o coração cheio de gratidão. Cada palavra escrita, cada noite de estudo e cada desafio superado foram impulsionados pelo amor e pela fé que me acompanharam até aqui. Sou imensamente grata a todos que, de alguma forma, estiveram comigo nessa trajetória.

Resumo

Este trabalho tem como objetivo desenvolver um modelo computacional baseado em Redes Neurais Artificiais (RNAs) para a triagem inicial do Transtorno do Espectro Autista (TEA) em adolescentes de 12 a 16 anos, resultando na criação do aplicativo *web* TRIA-TEA Adolescente. A pesquisa desenvolveu-se com estudo teórico sobre os principais conceitos relacionados ao TEA, suas manifestações na escola e a importância do diagnóstico precoce, seguido pela abordagem matemática e computacional que fundamenta o funcionamento das RNAs. O modelo proposto foi desenvolvido a partir de um conjunto de dados disponível no repositório *UCI Machine Learning Repository* da Universidade da Califórnia em Irvine, originalmente apresentado por Tabtah e posteriormente atualizado por Artoni *et al.*, contendo 248 amostras e 17 atributos de entrada. A Rede Neural implementada é do tipo MLP, composta por duas camadas ocultas contendo 128 e 64 neurônios com função de ativação ReLU, além de uma camada de saída contendo um único neurônio com a função sigmoide. O treinamento foi realizado com o otimizador Adam e função de perda de Erro Quadrático Médio (MSE), obtendo resultados satisfatórios de acurácia, precisão e sensibilidade. O modelo foi integrado a um aplicativo *web* acessível e gratuito, permitindo estimar a probabilidade de TEA de forma prática e intuitiva. Este trabalho evidencia o potencial das RNAs como ferramenta de apoio à inclusão e à educação, além de reforçar a importância da interdisciplinaridade entre Matemática, Ciência da Computação e Ciências Humanas.

Palavras-chave: : Redes Neurais Artificiais. Transtorno do Espectro Autista. Aprendizado de Máquina. Aplicativo. Triagem.

Abstract

This study aims to develop a computational model based on Artificial Neural Networks (ANNs) for the initial screening of Autism Spectrum Disorder (ASD) in adolescents aged 12 to 16, resulting in the creation of the TRIA-TEA Adolescente web application. The research was conducted through a theoretical study on the main concepts related to ASD, its manifestations in schools, and the importance of early diagnosis, followed by the mathematical and computational approach underlying the functioning of ANNs. The proposed model was developed using a dataset available in the UCI Machine Learning Repository of the University of California, Irvine, originally presented by Tabtah and later updated by Artoni et al., containing 248 samples and 17 input features. The implemented Neural Network is a Multilayer Perceptron (MLP) composed of two hidden layers with 128 and 64 neurons using the ReLU activation function, and an output layer with a single neuron employing the sigmoid function. Training was performed with the Adam optimizer and Mean Squared Error (MSE) loss function, achieving satisfactory results in accuracy, precision, and recall. The model was integrated into an accessible and free web application, allowing the probability of ASD to be estimated in a practical and intuitive manner. This work demonstrates the potential of ANNs as a support tool for inclusion and education, while also highlighting the importance of interdisciplinarity among Mathematics, Computer Science and Human Sciences.

Keywords: : Artificial Neural Networks. Autism Spectrum Disorder. Machine Learning. Application. Screening.

Lista de Abreviaturas

Abreviaturas utilizadas neste trabalho:

ABNT	Associação Brasileira de Normas Técnicas
CSA	Campus Santo Antônio
PROFMAT	Mestrado Profissional em Matemática em Rede Nacional
UFSJ	Universidade Federal de São João del Rei
TEA	Transtorno do Espectro Autista
RNA	Redes Neurais Artificiais
ONU	Organização das Nações Unidas
AEE	Atendimento Educacional Especializado
DSM-5	Diagnostic and Statistical Manual of Mental Disorders (Manual Diagnóstico e Estatístico de Transtornos Mentais)
CIPTEA	Carteira de Identificação da Pessoa com Transtorno do Espectro Autista
SEDESE-MG	Secretaria de Estado de Desenvolvimento Social de Minas Gerais
IBGE	Instituto Brasileiro de Geografia e Estatística
TDAH	Transtorno de Déficit de Atenção e Hiperatividade
IA	Inteligência Artificial
MLP	<i>Perceptron</i> Multicamadas
MSE	Erro Quadrático Médio
Tanh	Função Tangente Hiperbólica
SGD	<i>Stochastic Gradient Descent</i>
GPU	Unidade de Processamento Gráfico
Adam	<i>Adaptive Moment Estimation</i>
VP	Verdadeiro positivo
VN	Verdadeiro negativo
FP	Falso positivo
FN	Falso negativo

Lista de Figuras

2.1	Símbolo do TEA.	18
2.2	Modelo de CIPTEA.	18
2.3	Relação percentual de estudantes brasileiros com 6 anos ou mais enfatizando sexo e diagnóstico de TEA.	25
4.1	RNA <i>Perceptron</i> de quatro camadas	36
4.2	Representação de um <i>overfitting</i>	40
4.3	Neurônios e conexões de uma RNA.	41
4.4	Comportamento da Função Sigmoide	45
4.5	Comportamento da Função ReLU	46
4.6	Comportamento da Função Tangente Hiperbólica	47
4.7	Comportamento da Função Linear	48
4.8	Comportamento da Função <i>Softmax</i>	49
4.9	<i>Batch size</i> em um <i>dataset</i>	54
4.10	Influência da taxa de aprendizagem no processo de otimização.	57
5.1	Representação esquemática da RNA para conversão de temperaturas	63
5.2	Variação do erro com até 1000 épocas.	64
5.3	Variação do erro com até 100 épocas.	65
5.4	Arquitetura da RNA para prever a nota de um aluno	68
5.5	Arquitetura da RNA para previsão da nota de um aluno, exibindo os pesos (W) e <i>bias</i> (b) atribuídos de forma aleatória antes do processo de treinamento	69
5.6	Arquitetura da RNA com pesos e <i>bias</i> ajustados após o treinamento.	72
5.7	Evolução da Função Perda.	75
6.1	Fluxograma das etapas para elaboração do aplicativo baseado em RNAs.	76
6.2	Arquitetura da RNA usada para elaborar o aplicativo <i>web</i> Tria-TEA Adolescente.	80
6.3	Exemplo de resultado gerado pelo Tria-TEA Adolescente.	81
6.4	Página inicial do aplicativo <i>web</i> Tria-TEA Adolescente.	82
6.5	Diagrama de navegação do aplicativo proposto (TRIA-TEA Adolescente).	83

Lista de Tabelas

4.1	Classificação dos elementos de uma Rede Neural Artificial	38
4.2	Principais funções de ativação e seus tipos de aplicação	50
5.1	Temperaturas utilizadas para treinamento da RNA.	62
5.2	<i>Dataset</i> utilizado para treinamento e teste da RNA	67
5.3	Descrição das colunas do <i>dataset</i>	67
5.4	Cálculos manuais da primeira época da RNA.	70
5.5	Cálculo manual do erro MSE na primeira época	71
5.6	<i>Dataset</i> para treinar a RNA sobre Intervenção Pedagógica.	74
6.1	Perguntas correspondentes aos 17 atributos do conjunto de dados de triagem.	78

Sumário

1	INTRODUÇÃO	12
2	Transtorno do Espectro Autista	17
2.1	Os sinais mais comuns do TEA	19
2.2	Causas do autismo	20
2.3	Tratamentos	22
2.4	Diagnóstico	23
2.5	Autismo na escola: Relato de experiência	26
3	Conceitos Básicos da Matemática do Ensino Superior	29
3.1	Preliminares do Cálculo de Funções de duas Variáveis	29
3.2	Noções de Álgebra Linear	33
4	Redes Neurais Artificiais: estudo teórico	35
4.1	Componentes de uma RNA	38
4.1.1	Camada de Entrada (<i>Input</i>)	39
4.1.2	Camadas Ocultas (<i>Hidden Layers</i>)	39
4.1.3	Camada de Saída (<i>Output</i>)	40
4.1.4	Neurônios	40
4.1.5	Atributos	42
4.2	Parâmetros	42
4.2.1	Pesos	42
4.2.2	<i>Bias</i>	42
4.3	Hiperparâmetros	43
4.3.1	Função de Ativação	43
4.3.2	Função de Custo (<i>Loss</i>)	50
4.3.3	Épocas (<i>epochs</i>)	52
4.3.4	<i>Batch Size</i>	53
4.3.5	Otimizadores e Taxa de Aprendizagem	55
4.3.6	<i>Momentum</i>	57
4.4	Aplicação de RNAs na resolução de problemas	58
4.4.1	Treinamento	58
4.4.2	Teste	58
4.4.3	Aplicação	60
5	Redes Neurais Artificiais: aplicações básicas	61
5.1	Modelo de RNA para conversão de temperaturas: Celsius para Fahrenheit	61
5.2	Modelo de RNA para prever a nota de um aluno	66
5.3	Modelo de RNA para identificar alunos que realizarão Intervenção Pedagógica	73

6	Modelo de RNA Proposto e Desenvolvimento do Aplicativo	76
6.1	Coleta de Dados	76
6.2	Pré-processamento	77
6.3	Treinamento e Arquitetura do Modelo	79
6.4	Pós-processamento e Integração	80
6.4.1	O Aplicativo Web TRIA-TEA Adolescente	82
6.5	Formulação Matemática do Modelo	84
7	CONSIDERAÇÕES FINAIS	88
	REFERÊNCIAS	90
	APÊNDICES	
A	Códigos dos Modelos de RNA	96
A.1	Modelo de RNA para conversão de temperaturas	96
A.2	Modelo de RNA para prever notas de alunos	99
A.3	Modelo de RNA para identificar alunos que realizarão Intervenção Pedagógica	102

1 INTRODUÇÃO

Nas últimas décadas, a humanidade tem presenciado uma transformação sem precedentes impulsionada pela tecnologia digital. Rodrigues-Jr., Oliveira e Oliveira Jr. (2021) destacam que a velocidade dos avanços atuais é muito maior do que em qualquer outra época, sendo impulsionada, principalmente, pela Inteligência Artificial (IA). Cozman e Neri (2021) definem IA como:

“[...] computadores digitais cujos programas representam e raciocinam sobre conhecimento e crenças, tomam decisões e aprendem, e interagem com seu ambiente, realizando todas essas atividades ou pelo menos algumas com nível alto de sofisticação.”

A IA tem se consolidado como uma das tecnologias mais transformadoras do século XXI, impactando profundamente diferentes setores da sociedade. Seu uso em campos diversos, tais como engenharia, economia, agronomia e medicina, tem demonstrado grande potencial para otimizar processos, realizar previsões, extrair características e efetuar classificações, entre outras (FURTADO, 2019). Essa presença crescente reforça a importância de compreender seus fundamentos e aplicações, especialmente em contextos voltados à melhoria da qualidade de vida das pessoas.

De acordo com Rodrigues-Jr., Oliveira e Oliveira Jr. (2021), essa tecnologia, alimentada por grandes volumes de dados, tem permitido que computadores realizem tarefas antes restritas a especialistas humanos, como interpretar imagens ou gerar textos. Segundo esses autores, embora seja improvável que as máquinas substituam completamente os humanos em atividades de natureza mais complexa, como o ensino em sala de aula, é esperado que aqueles que souberem utilizar os recursos da IA de forma eficiente se destaquem em relação aos que não o fizeram.

Sistemas computacionais dotados de algum tipo de inteligência têm sido aplicados nas mais diversas áreas do conhecimento (RODRIGUES-Jr., OLIVEIRA e OLIVEIRA Jr., 2021). Segundo Carvalho (2024), o avanço científico e o aumento do poder de

processamento dos computadores possibilitaram o surgimento de métodos matemáticos que impulsionaram a área da computação. Além disso, conforme essa autora, o fascínio pela complexidade e pela capacidade do cérebro de aprender e se adaptar tem inspirado diversos estudos, permitindo o desenvolvimento de conceitos como as Redes Neurais Artificiais (RNAs). Essas redes, responsáveis por avanços recentes em aprendizado de máquina e reconhecimento de padrões, buscam imitar, de maneira simplificada, o processo de aprendizagem do cérebro humano.

As RNAs podem ser organizadas de diversas formas, de acordo com a maneira como suas camadas e conexões são estruturadas. Dentre essas, destacam-se as redes do tipo *feedforward*, nas quais o fluxo de informação ocorre apenas em um único sentido (da camada de entrada para a de saída), sem conexões de retorno entre os neurônios (FURTADO, 2019). Este trabalho faz uso das Redes *Perceptron* Multicamadas (*Multilayer Perceptron*), que representam uma subclasse das redes *feedforward* e são amplamente empregadas em tarefas de classificação e regressão devido à sua capacidade de modelar relações não lineares complexas (CARVALHO, 2024).

As RNAs são modelos matemáticos inspirados no funcionamento do sistema nervoso central, buscando reproduzir a forma como os neurônios processam e transmitem informações (BRASIL *et al.*). Segundo Mazzon e Barros (2021), técnicas de modelagem quantitativa, como as RNAs, são amplamente utilizadas por cientistas de dados para resolver diferentes tipos de problemas, de acordo com a natureza dos dados e o objetivo da modelagem. Nos problemas de regressão, a saída corresponde a um valor numérico contínuo, como, por exemplo, a previsão da temperatura ou da nota de um aluno. Por outro lado, nos problemas de classificação, o modelo deve atribuir cada amostra a uma entre as classes possíveis, como identificar quais alunos farão prova de Intervenção Pedagógica ou identificar se uma pessoa possui ou não Transtorno do Espectro Autista (TEA). Os exemplos apresentados não foram escolhidos ao acaso: eles correspondem exatamente aos modelos de RNAs abordados ao longo deste trabalho, sendo o último exemplo, inclusive, o objetivo principal desta pesquisa.

O TEA é uma condição complexa do neurodesenvolvimento cujo diagnóstico ainda apresenta desafios. Segundo Silva e Fernandes (2023), mesmo com os avanços científicos, o diagnóstico continua sendo impreciso, e exames genéticos ainda não são capazes de confirmar com exatidão a presença desse transtorno. De acordo com Nicoletti e Honda

(2021), o diagnóstico do TEA é essencialmente clínico e depende da análise de diversos aspectos, como a comunicação verbal e não verbal, o nível intelectual, o contexto familiar e educacional e o grau de autonomia do indivíduo. Mangabeira (2022) ressalta que no Brasil o diagnóstico ainda ocorre tardiamente, especialmente nos casos mais leves, e o diagnóstico precoce é fundamental para um melhor prognóstico.

Quirino *et al.* (2025) afirmam que:

“[...] uma triagem adequada viabiliza o diagnóstico precoce e favorece um manejo eficaz, melhorando o desenvolvimento e a qualidade de vida do paciente.”

Mas, de acordo com esses autores, ainda existem diversas limitações relacionadas à triagem e ao diagnóstico. Uma delas refere-se ao fato do acesso a especialistas ser demorado. Outra limitação diz respeito à escassez de instrumentos voltados para pacientes brasileiros. Por isso, é essencial o uso de instrumentos de triagem padronizados, acessíveis e abrangentes.

Este trabalho apresenta o desenvolvimento de um aplicativo para triagem inicial de TEA. A triagem inicial é feita de forma individual para cada faixa etária, e a escolhida foi adolescentes de 12 a 16 anos, um período que, segundo Mutschler e Silva (2025):

“configura-se como uma fase complexa, desafiadora e repleta de potencialidades, na qual a preservação da autoestima e da segurança, tanto individual quanto coletiva, constitui um alicerce fundamental para o desenvolvimento do jovem.”

Além disso, é justamente a faixa etária dos alunos do Ensino Fundamental II e Ensino Médio.

O aplicativo criado para triagem inicial de TEA em adolescentes é baseado em um modelo de RNA *Perceptron* Multicamadas. Essa proposta visa explorar o potencial das redes neurais na identificação de padrões comportamentais e cognitivos associados ao transtorno, oferecendo uma solução acessível e de fácil utilização. Além disso, busca-se evidenciar a aplicação prática de conceitos matemáticos e computacionais abordados ao longo do curso, especialmente no contexto do ensino e da modelagem.

Para a elaboração das RNAs apresentadas neste trabalho, utilizou-se a linguagem *Python*. Segundo Feitosa (2024), essa linguagem é uma das mais populares e amplamente utilizadas no mundo atualmente. O autor destaca diversas razões pelas quais o *Python* é uma escolha apropriada para análise de dados e desenvolvimento de modelos de IA, entre elas:

- Facilidade de aprendizado tanto para programadores experientes quanto para iniciantes;
- Ecossistema robusto de bibliotecas e *frameworks* que facilitam o desenvolvimento de soluções de análise de dados e IA;
- Ampla comunidade de suporte, com recursos como tutoriais, fóruns de discussão e projetos de código aberto;
- Fácil integração com outras linguagens e ferramentas;
- Suporte para computação científica e estatística;
- Aplicações em diversos domínios.

Motivados pelas vantagens mencionadas, optou-se por trabalhar com a linguagem de programação *Python*, uma vez que ela se mostra adequada para a implementação e experimentação de modelos de RNAs, foco deste estudo.

O trabalho está estruturado da seguinte maneira: o Capítulo 2 apresenta uma revisão teórica sobre o TEA, destacando suas principais características, tratamentos e desafios relacionados ao diagnóstico. O Capítulo 3 aborda os fundamentos matemáticos básicos e preliminares necessários para a compreensão do funcionamento das RNAs, com ênfase em conceitos como mínimos e gradiente descendente. No Capítulo 4, são descritos a estrutura e o funcionamento das RNAs, incluindo suas principais componentes, parâmetros, hiperparâmetros, processos de aprendizagem e aplicação. O Capítulo 5 apresenta três aplicações básicas de RNAs abordando exemplos didáticos. O Capítulo 6 descreve o processo de criação do modelo de RNA voltado à triagem de TEA em adolescentes e o desenvolvimento do aplicativo correspondente. Por fim, o Capítulo 7 traz as considerações finais, destacando as contribuições do trabalho e possíveis extensões futuras.

Além de seu valor prático, o desenvolvimento deste projeto contribui para a formação científica e tecnológica, promovendo a integração entre matemática, tecnologia, educação inclusiva e saúde. Ademais, incentiva o raciocínio lógico, a experimentação e a aplicação de conhecimentos teóricos em situações reais. Espera-se que este estudo contribua não apenas para a compreensão teórica e prática das RNAs, mas também para demonstrar como ferramentas matemáticas e computacionais podem ser aplicadas de forma interdisciplinar,

promovendo avanços tanto na pesquisa quanto no ensino e na saúde. Além disso, busca-se favorecer, de alguma forma, o diagnóstico precoce do TEA a partir de uma triagem efetiva.

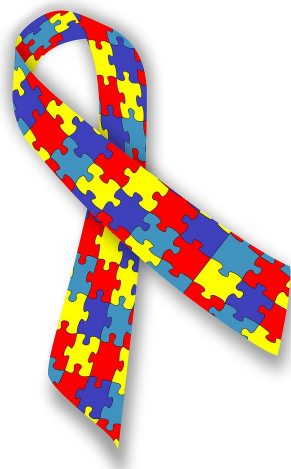
2 Transtorno do Espectro Autista

O estudo sobre o TEA não é recente. Segundo Silva e Fernandes (2023), a palavra “Autismo” apareceu pela primeira vez em 1906, quando Eugen Bleuler (Suíça) a usou para descrever certos sintomas como a “tendência a se desconectar da interação com o meio ambiente”. Nesse período o autismo ainda era tratado como esquizofrenia. Na década de 1940, o médico Leo Kanner (Áustria-Ucrânia), ajudou a entendê-lo melhor, separando autismo da esquizofrenia e definindo aspectos do autismo como tendência à solidão e necessidade de rotina (EVÊNCIO e FERNANDES, 2019).

Posteriormente, a pesquisa em torno do TEA começou a ser intensificada. Com o passar dos anos, esse assunto foi ganhando relevância devido aos muitos estudos, grupos e o interesse da mídia. A Organização das Nações Unidas (ONU) decretou em 2007 o dia 02 de abril como marco para o Dia Mundial da Conscientização do Autismo e em 2018 essa data passou a fazer parte também do calendário brasileiro, um modo de pautar o assunto e dar sua devida importância (SILVA e FERNANDES, 2023). Ainda de acordo com Silva e Fernandes, em 2012 no Brasil, foi implantada a Lei nº 12.764, que instituiu a Política Nacional de Proteção dos Direitos da Pessoa com TEA, definindo que pessoas com TEA são consideradas pessoas com deficiência para todos os efeitos legais.

Devido a diversidade dos casos e a complexidade de diagnóstico, a fita de quebra-cabeças, representada na Figura 2.1, com diferentes cores e formas foi adotada como símbolo do TEA. A Lei nº 13.977, determina que os estabelecimentos públicos e privados possam utilizar a fita de quebra-cabeças para identificar que indivíduos com TEA tenham preferência no atendimento.

Figura 2.1: Símbolo do TEA.



Fonte: JÚNIOR, 2010.

A Lei nº 13.977 também prevê a criação da Carteira de Identificação da Pessoa com Transtorno do Espectro Autista (CIPTEA). De acordo com Silva e Fernandes (2023), esse documento assegura prioridade ao autista nas áreas de saúde, educação e assistência social. Ressalta-se que a emissão da CIPTEA é gratuita. A seguir, na Figura 2.2, apresenta-se um modelo da carteira.

Figura 2.2: Modelo de CIPTEA.

ciptea		Carteira de Identificação da Pessoa com Transtorno do Espectro Autista	
Nome: MIGUEL ALVES BARBOSA TONELLI		DADOS DE CONTATO DA PESSOA COM AUTISMO	
Nome da Mãe: MARIANA PACHECO VIEIRA		Endereço: RUA LOGRADOURO, 123, FLORESTA, MATOZINHOS / MG - 35.720-000	
CPF: 000.000.00-00		Telefone: (31) 99999-9999	
Tipo Sanguíneo: A+		DADOS DO RESPONSÁVEL OU CUIDADOR	
Naturalidade: MG/MATOZINHOS		Nome: FELIPE ALVES BARBOSA TONELLI	
CONTATO DE EMERGÊNCIA: (31) 99999-9999 MARINA FERREIRA DA ROCHA		Telefone: (31) 99999-9999	RG: 00.000.000-0
Atendimento Preferencial Lei Nº 13.977/2020		Número Ciptea: #####	
Válido em Todo Território Nacional		Emissão: 04/01/2022	
XXXXXXXXXX		Validade: 04/01/2027	
		Observações:	
		Assinatura ou Digital	
		SECRETARIA DE ESTADO DE DESENVOLVIMENTO SOCIAL Romeu Zema Neto Governador Elizabeth Jucá e Mello Jacometti Secretária de Secretaria de Estado de Desenvolvimento Social (SEDESE)	

Fonte: SEDESE-MG.

Para conhecer um pouco mais sobre o TEA, abordaremos alguns aspectos importantes a seguir.

2.1 Os sinais mais comuns do TEA

Os sintomas do TEA se fazem presentes desde cedo, e essas características já exercem impacto no cotidiano das famílias e nas relações entre seus membros (SILVA e FERNANDES, 2023). Por ser um transtorno com ampla variabilidade (espectro) e afetar de forma diferente cada indivíduo, o TEA tem sinais/sintomas mais comuns. Mangabeira (2022) destaca como as principais características do TEA:

- Apresentar atraso anormal na fala;
- Contato visual reduzido e desinteresse em interagir com pessoas e objetos ao redor, inclusive quando for chamado;
- Dificuldade de interpretar gestos e expressões faciais;
- Preferência por brincar sozinho e dificuldade de participar de atividades em grupo;
- Sensibilidade a ruídos;
- Dificuldade para formular frases ou repetição frequente da mesma frase ou palavra (ecolalia);
- Seletividade alimentar, em relação a cheiro, sabor e textura;
- Apresentar movimentos repetitivos, rígidos ou estereotipados durante as interações;
- Mostrar interesse limitado e fixo por objetos ou assuntos considerados incomuns ou excêntricos, como biologia, paleontologia, tecnologia, datas, números, entre outros;
- Regressão gradual em habilidades sociais e cognitivas;
- Hiperatividade e déficit de atenção;
- Falta de empatia diante de determinadas situações;
- Aumento ou redução na resposta à dor e à temperatura.

Além disso, de acordo com Evêncio e Fernandes (2019), por se manifestarem em diferentes níveis de intensidade, uma pessoa com TEA pode ser classificada de acordo com a dependência e necessidade de apoio exigido para realizar atividades do cotidiano.

- Uma pessoa diagnosticada como de **grau 1 de suporte (antiga síndrome de Asperger)** apresenta prejuízos leves, que podem não a impedir de estudar, trabalhar e se relacionar.
- Uma pessoa com **grau 2 de suporte** tem um menor grau de independência e necessita de algum auxílio para desempenhar funções cotidianas, como tomar banho ou preparar a sua refeição.
- Uma pessoa com **grau 3 de suporte** vai manifestar dificuldades graves e costuma precisar de apoio especializado ao longo da vida.

É possível que a pessoa transite por esses diferentes níveis, ou seja, a pessoa pode mudar o seu nível de suporte ao longo da vida.

2.2 Causas do autismo

O TEA é um distúrbio caracterizado pela alteração do desenvolvimento neurológico do indivíduo, tendo efeito na dificuldade no convívio social, na comunicação verbal e não verbal e em interesses específicos por algumas atividades realizadas de forma repetitiva (NICOLETTI e HONDA, 2021). Mas saber o que acontece com o cérebro dessas pessoas ainda é um mistério para a ciência. Por isso, as causas do TEA não são totalmente conhecidas. Acredita-se que seja multifatorial, associada a uma combinação de influências genéticas, biológicas e ambientais.

Quirino *et al.* afirmam que a influência genética é alta, visto que, a herdabilidade é pode chegar até a 90%. Entretanto, Lavor *et al.* (2021), afirmam que alterações genéticas associadas ao autismo também podem ser encontradas em pessoas que não apresentam a condição. Além disso, segundo esses mesmos autores, são considerados biomarcadores do TEA: a grande diferença de expressão do ácido ribonucleico entre o córtex frontal e o temporal, em comparação com o cérebro controle, e as atividades anormais das vias de sinalização de cálcio e das proteínas quinases ativadas por mitógeno.

Não obstante, os fatores ambientais também são estudados e comprovados. Segundo Silva e Fernandes(2023), na Suécia, em 2014, foi realizado o maior estudo sobre as causas do autismo e concluíram que fatores ambientais são tão importantes quanto a genética para desenvolver TEA. Alguns desses fatores ambientais surgem já na gestação da pessoa com TEA, como estresse, infecções, consumo/exposição a substâncias tóxicas, desequilíbrios

metabólicos e complicações durante a gravidez.

Por sua vez, Costa *et al.* (2022) destacam que fatores não genéticos também podem influenciar o desenvolvimento do TEA. Entre eles, a exposição do feto a medicamentos utilizados pela mãe durante a gestação pode afetar o desenvolvimento cognitivo e comportamental da criança. Essas autoras também afirmam que: “estudos indicam associação do TEA com o uso de analgésicos, antitérmicos, antidepressivos, antiepiléticos, anti-asmáticos, antidiabéticos e antibióticos”.

Da mesma forma, Costa *et al.* (2022) afirmam que o consumo de tabaco, álcool ou drogas lícitas e ilícitas durante a gestação também pode aumentar o risco de TEA. O tabaco, por exemplo, pode comprometer o desenvolvimento cerebral e a circulação placentária, enquanto álcool e drogas ilícitas afetam diretamente o neurodesenvolvimento. Além disso, Maia (2022) cita que:

“Vale ressaltar que os fatores ambientais contribuem para determinar o risco individual ou a gravidade do TEA; a exposição a produtos químicos ambientais pode alterar a metilação do DNA em múltiplos tecidos, incluindo o cérebro e que mudanças na metilação do DNA foram documentadas em indivíduos com o TEA e implicadas na patogênese desse transtorno.”

Oliveira, Silveira e Maia (2022) também afirmam que crianças/adolescentes com TEA apresentaram associação com complicações no pós-natal, como má formação genética, icterícia neonatal, ausência de choro ao nascer e episódios de convulsão na infância. Essas autoras indicam que a presença de má formação congênita ou genética aumenta o risco de desenvolvimento do TEA, embora as causas exatas ainda não tenham sido determinadas, sendo provável a influência de fatores genéticos e ambientais subjacentes. Quanto à icterícia neonatal, essas autoras afirmam que crianças que desenvolvem TEA são mais propensas a apresentá-la e, em alguns casos, necessitam de tratamento devido ao potencial efeito tóxico da bilirrubina sobre o sistema nervoso central, o que pode resultar em sequelas cerebrais. Sobre a ausência de choro ao nascer, embora nem sempre tenha sido associada ao TEA, pode estar relacionada a distúrbios respiratórios neonatais, como hipóxia, que podem afetar o desenvolvimento cerebral. Por fim, episódios de convulsão na infância podem gerar alterações estruturais no córtex pré-frontal, interferindo na função cerebral e contribuindo, a longo prazo, para o surgimento de distúrbios comportamentais, incluindo o autismo, o TDAH e deficiências intelectuais mais graves.

Além disso, nas últimas décadas, segundo o estudo feito por Maia *et al.*(2018), observa-se a associação com o TEA foi maior quando os genitores apresentavam idades avançadas. Sendo essa uma causa do TEA, ela também está atrelada ao aumento no número de casos de TEA, visto que casais estão tendo filhos em idades mais avançadas, esperando, muitas vezes, obter êxito profissional para depois terem filhos.

Vale ressaltar que, nos anos 2000, a mídia chegou a associar o aumento dos casos de TEA à vacina tríplice viral (Sarampo, Caxumba e Rubéola) e que, mais recentemente, algumas pessoas chegaram a relacionar o TEA às vacinas contra a COVID-19 (SILVA e FERNANDES, 2023). No entanto, tais alegações foram amplamente refutadas por estudos científicos.

Outro fator ambiental que está sendo discutido atualmente é sobre o uso excessivo de telas por crianças. Com o aumento das tecnologias acessíveis à população em geral, o uso de telas por crianças precocemente e de forma excessiva pode estar relacionado ao TEA. O estudo de Morais *et al.*(2025) menciona que:

“[...] embora não haja consenso sobre o tempo de tela como fator causal do TEA, há evidências de que sua exposição prolongada pode agravar sintomas em crianças predispostas, especialmente no que se refere a déficits na comunicação, dificuldades na interação social e padrões de comportamento repetitivos.”

Portanto, as causas para o autismo são ainda estudadas. Mas há entre elas, fatores genéticos, biológicos e ambientais.

2.3 Tratamentos

Até o momento, O TEA não possui cura conhecida, e devido a cada caso ser singular, com características próprias individualizadas, cada paciente exige um tipo de acompanhamento específico. Com isso, é necessária a participação dos familiares e de uma equipe de diferentes profissionais, como mencionados por Silva e Fernandes (2023) e Maia *et al.* (2025):

- Médicos: para acompanhar as evoluções e desenvolvimentos da pessoa autista. Embora não existam medicamentos para o TEA, alguns remédios podem ser receitados pelo médico para ajudar com possíveis sintomas como ansiedade, agressividade, hiperatividade e alterações de humor;

- Fisioterapeutas: para ajudar a desenvolver habilidades motoras, melhorar a coordenação, o equilíbrio e a consciência corporal;
- Fonoaudiólogos: para orientar no desenvolvimento verbal e gestual;
- Terapeutas Ocupacionais: para desenvolver a parte motora e simular interações e movimentos do cotidiano;
- Psicólogo: para estimular condutas positivas, reduzir comportamentos nocivos e cuidar da saúde emocional. Vale mencionar o modelo de tratamento que, segundo Silva e Fernandes (2023) mais apresenta resultados, a Terapia Cognitivo Comportamental. Essa terapia apresenta um conjunto de técnicas para oportunizar a adaptação comportamental da pessoa com TEA ao ambiente social em que está inserida (como a escola, por exemplo) e tem apoio e participação ativa da família;
- Educadores: para acompanhar o desenvolvimento escolar, incluindo efetivamente o aluno com TEA, adaptando o que for necessário para que ele esteja inserido na escola e tenha educação de qualidade.

Os tratamentos mencionados são usados para incentivar o indivíduo a desenvolver autonomia, realizar tarefas cotidianas, se comunicar e conviver em sociedade, ter maior estabilidade emocional e assim, qualidade de vida. Além disso, devem ser intensivos e duradouros.

2.4 Diagnóstico

Segundo Silva e Fernandes (2023), o Manual Diagnóstico e Estatístico de Transtornos Mentais (DSM-5) afirma que “o diagnóstico de TEA é baseado em duas áreas: (I) prejuízo persistente na interação e na comunicação social e (II) padrões restritos e repetitivos de comportamento, interesses ou atividades”. Os principais sintomas já aparecem nos primeiros meses de vida. Apesar de familiares terem suspeitas, é importante salientar que apenas médicos podem diagnosticar doenças, indicar tratamentos e receitar remédios. O diagnóstico acontece através de uma entrevista/questionários preenchidos, da observação e da história clínica (MANGABEIRA, 2022).

Ainda de acordo com Silva e Fernandes (2023), em 2021, estimativas apontaram que 1 a cada 44 crianças nascidas nos Estados Unidos da América eram autistas. Em

2019, de acordo com esses autores, a partir da Lei 13.861, tornou-se obrigatório a inclusão de dados e informações específicas sobre TEA nos censos demográficos no Brasil. Esta estatística é extremamente importante, como afirmam Nicoletti e Honda (2021):

“Esta é uma Lei que contribui de forma significativa para familiares e indivíduos com TEA, pois torna estas pessoas visíveis para os governos federal, estadual e municipal. Além disso, quando houver a contagem destes indivíduos e familiares, isso facilitará que os governos desenvolvam políticas públicas mais efetivas permitindo uma maior inclusão dos indivíduos com TEA na sociedade, além de promover maior visibilidade desta condição.”

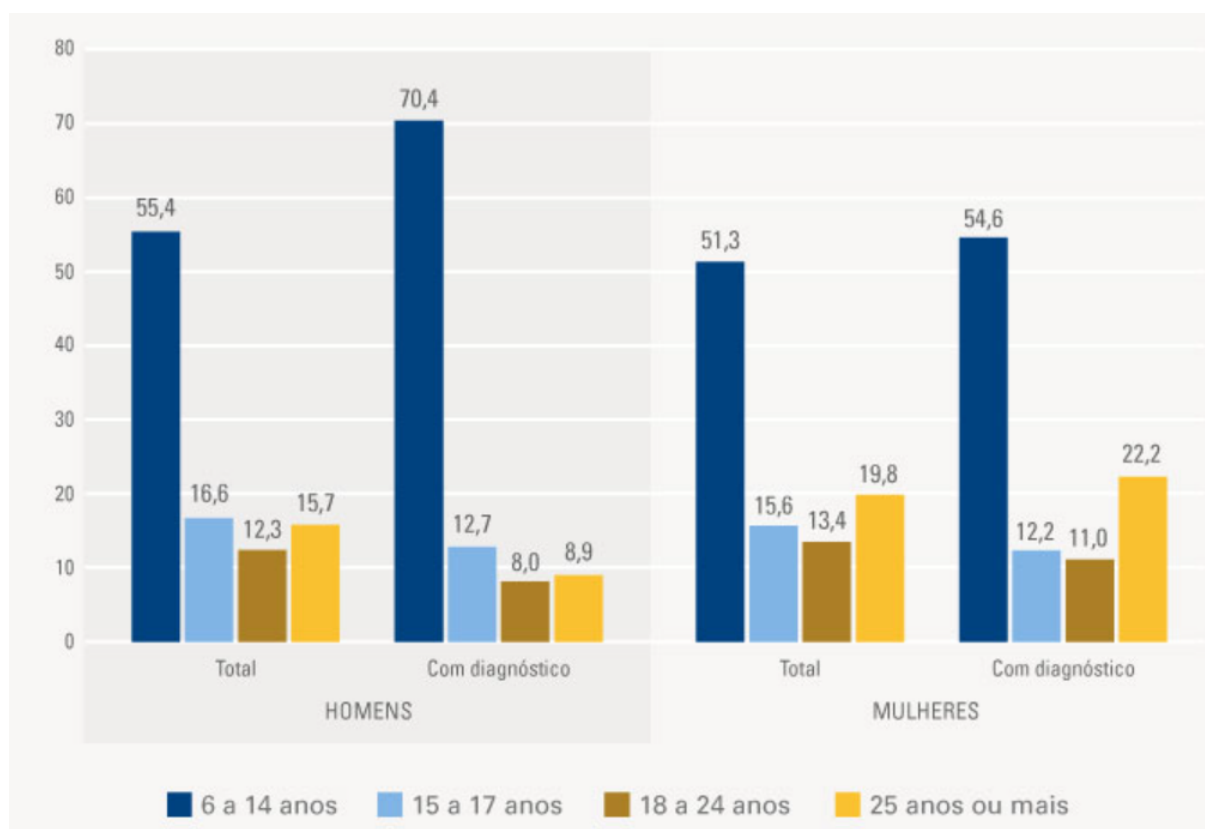
As informações do “Censo Demográfico 2022: Pessoas com Deficiência e Pessoas Diagnosticadas com TEA - Resultados preliminares da amostra”, divulgado pelo IBGE (Instituto Brasileiro de Geografia e Estatística) no dia 23 de maio de 2025 trouxe pela primeira vez estatísticas sobre o TEA no Brasil. A pesquisa foi realizada com uma amostra da população e quem a respondeu declarou se os moradores do domicílio já haviam sido diagnosticados com TEA por algum profissional de saúde. Esse Censo Demográfico identificou:

- 2,4 milhões de pessoas com diagnóstico de TEA, o que corresponde a 1,2% da população brasileira.
- A prevalência foi maior entre os homens (1,5%) do que entre as mulheres (0,9%) o que corresponde a 1,4 milhões de homens e 1,0 milhão de mulheres foram diagnosticados com autismo por algum profissional de saúde.
- Entre os grupos etários, a prevalência de diagnóstico de autismo foi maior entre os mais jovens: 2,1% no grupo de 0 e 4 anos de idade, 2,6% entre 5 e 9 anos, 1,9% entre 10 e 14 anos e 1,3% entre 15 e 19 anos. Esses percentuais representam, ao todo, 1,1 milhão de pessoas de 0 a 14 anos com autismo. Nos demais grupos etários, os percentuais oscilaram entre 0,8% e 1,0%.
- Na desagregação por cor ou raça, o maior percentual de pessoas com autismo se deu entre as pessoas declaradas brancas, com 1,3% o que equivale a 1,1 milhão de pessoas. A menor prevalência está entre as pessoas de cor ou raça indígena, com 0,9% o que representa 11,4 mil pessoas.

- A taxa de escolarização da população com autismo (36,9%) foi superior à observada na população geral (24,3%). Essa diferença foi mais expressiva entre os homens: 44,2% dos homens com autismo estavam estudando, frente a 24,7% do total. Entre as mulheres, a taxa de escolarização foi 26,9% entre aquelas com autismo, ante 24,0% no total.

A Figura 2.3 mostra a relação percentual de estudantes com 6 anos ou mais por grupos de idade, sexo e existência de diagnóstico de TEA. Nesse gráfico, observa-se que a prevalência do TEA é maior entre os homens do que entre as mulheres, como já havia sido mencionado. Além disso, a faixa etária com maior concentração de diagnósticos situa-se entre 5 e 9 anos. Esses dados evidenciam tanto a diferença na distribuição do TEA entre os sexos quanto a importância do período da infância para a identificação e diagnóstico do transtorno (IBGE, 2025).

Figura 2.3: Relação percentual de estudantes brasileiros com 6 anos ou mais enfatizando sexo e diagnóstico de TEA.



Fonte: Censo Demográfico 2022.

Apesar do avanço de, pela primeira vez, termos dados estatísticos sobre o TEA no Brasil, é importante destacar que a pesquisa questionava sobre casos que já possuíam

diagnósticos. Entretanto, muitos dos casos ainda não foram diagnosticados. Assim, o Brasil ainda tem muito o que se fazer principalmente para tornar o diagnóstico do TEA mais abrangente, prévio e eficaz.

2.5 Autismo na escola: Relato de experiência

De acordo com Quirino *et al.* (2025), nos últimos anos, a incidência do TEA aumentou consideravelmente. Devido a esse aumento, o número de crianças autistas na Educação Básica também tem crescido. Por isso, é de grande importância a realização de pesquisas, tanto para possibilitar o diagnóstico precoce (permitindo o início antecipado dos tratamentos), quanto na área educacional, voltadas à capacitação de professores e gestores escolares para promover uma educação mais inclusiva (BANDEIRA, 2022).

A escola ocupa um papel central no processo de identificação de possíveis alterações no desenvolvimento infantojuvenil, uma vez que, anualmente, recebe novos alunos e passa a acompanhar de forma contínua seu desempenho acadêmico, comportamento social e interações em sala de aula. Nesse contexto, os professores, por estarem em contato direto e frequente com os estudantes, tornam-se agentes fundamentais na observação de comportamentos diferenciados, inclusive aqueles mais sutis, que podem indicar a necessidade de acompanhamento especializado.

De acordo com Da Silva e De Souza (2023), a atuação conjunta entre professores, coordenação pedagógica e família é essencial para que sejam realizados encaminhamentos adequados aos profissionais da saúde, possibilitando melhores resultados tanto no âmbito escolar quanto no convívio familiar e social da criança. A atenção precoce e o acompanhamento interdisciplinar contribuem significativamente para o desenvolvimento do aluno, favorecendo sua participação nas atividades escolares e promovendo maior inclusão no ambiente educacional.

Para ilustrar como as características do TEA se manifestam na prática escolar, apresento algumas experiências vivenciadas enquanto professora na Educação Básica. Esses exemplos não têm caráter estatístico, mas permitem compreender a aplicação dos conceitos teóricos abordados anteriormente.

Durante minha atuação como professora da Educação Básica em uma escola pública, tive alguns alunos com laudo de TEA. E, como mencionado por Silva e Fernandes (2013), é frequente a ocorrência de mais de um transtorno do neurodesenvolvimento. Os alunos

que tive com laudo de TEA também não continham apenas TEA, mas também outras comorbidades como dislexia, transtorno de ansiedade, deficiência intelectual e síndrome do X frágil.

Um desses alunos acompanhava o ensino regular e, inclusive, tinha facilidade em realizar cálculos e interpretar problemas. Entretanto, no 6º ano do Ensino Fundamental ainda estava sendo alfabetizado devido ao quadro de dislexia e por isso era acompanhado por um professor de Atendimento Educacional Especializado (AEE). Ele interagia bem com os colegas e professores, mas ficava ansioso durante a realização de provas e atividades avaliativas. Esse aluno demonstrava o nervosismo com choro e movimentos repetitivos com as mãos durante a realização das mesmas. Nas atividades extraclasse também era notório um certo desconforto do aluno com TEA, diferentemente dos demais alunos da turma, que demonstravam animação em sair da rotina escolar.

Outra aluna com laudo de TEA que tive também o prazer de lecionar tinha um grau de suporte maior. Ela apresentava algumas características típicas do TEA, como: dificuldades na fala, principalmente a formular frases e dificuldade em interagir com os colegas, além de ser sensível a ruídos. Especificamente em matemática, ela estava conhecendo os números maiores que dez no 7º ano do Ensino Fundamental. Tinha também dispraxia, caracterizada por dificuldades motoras.

Além dos alunos com diagnóstico médico confirmado de TEA, tive também outros estudantes sem laudo médico, mas que apresentavam sinais característicos do transtorno. Em um caso, a família de um desses alunos me procurou enquanto eu atuava como professora, solicitando auxílio para responder um questionário aplicado por uma neuropsicóloga como parte do processo de avaliação e obtenção de diagnóstico, suspeitando que o estudante pudesse estar dentro do espectro autista.

Com base em minhas experiências, observa-se que o TEA é um espectro no qual cada pessoa autista possui sua própria individualidade. Há alunos com TEA que não estudam o mesmo que os alunos da turma regular, outros acompanham. Alguns têm dificuldades, outros facilidade. É relativo. Por isso, nem sempre é necessário adaptar o conteúdo ou planejar aulas diferentes quando se tem alunos com TEA. Bandeira (2022) afirma que:

“Apesar do TEA não ser um transtorno específico de aprendizagem, as respostas atípicas aos estímulos ambientais, alterações comportamentais e a forma como estabelecem o pensamento podem causar dificuldades no

processo de ensino e de aprendizagem. No entanto, o TEA se manifesta de formas diferentes em cada indivíduo, podendo ser classificado em graus, variando de 1 a 3 conforme a necessidade crescente de apoio para adaptação ao meio. Assim, as dificuldades na aprendizagem também ocorrem em graus variados.”

Portanto, como afirma Silva (2023), compreender e atender a diversidade de demandas exige um olhar sensível e uma abordagem personalizada.

Além disso, vale salientar que os meus alunos com TEA eram acolhidos pelos colegas, que demonstravam carinho e cuidado não apenas com eles, mas também com aqueles que apresentavam outras deficiências. Como afirma Silva (2023), um ambiente acolhedor e inclusivo requer o combate ao preconceito e a valorização de princípios como respeito, empatia e solidariedade. Tal postura é de extrema importância não apenas para que os alunos com TEA tenham seus direitos assegurados, mas também para que os demais aprendam a conviver em sociedade, reconhecendo e respeitando as diferenças, mesmo em momentos de crise. Como afirma Paulo Freire:

“A inclusão acontece quando se aprende com as diferenças e não com as igualdades”.

Diante do exposto, não é intenção deste trabalho criar um aplicativo para diagnosticar adolescentes, mas sim, propor um aplicativo para triagem inicial de TEA, que seja eficiente, usando as RNAs do tipo *perceptron* multicamadas adequadamente treinadas através da IA. Desenvolvido em português, o aplicativo visa facilitar o uso por pais, educadores e profissionais de saúde. A Matemática está por trás de todo o processo da criação do aplicativo, desde o desenvolvimento do questionário até a análise dos resultados. Através dos testes e da precisão do questionário, os professores podem utilizá-lo para fazer a triagem inicial ao encaminhar para o profissional da saúde especializado e os médicos, que são responsáveis por diagnosticar, podem usá-lo em suas consultas. Portanto, é fato que o aplicativo também foi pensado como uma ferramenta de inclusão.

3 Conceitos Básicos da Matemática do Ensino Superior

Os conceitos apresentados a seguir foram retirados e/ou adaptados de Stewart (2013) e Lima (2004, 2006, 2014). Não se trata de uma abordagem aprofundada, mas de um conjunto de definições e teoremas selecionados para oferecer o suporte teórico necessário à compreensão dos capítulos seguintes.

3.1 Preliminares do Cálculo de Funções de duas Variáveis

Definição 3.1 (Função Ímpar): Seja $D \subset \mathbb{R}$ e $f : D \rightarrow \mathbb{R}$ uma função real de variável real. f é chamada **função ímpar** se $f(-x) = -f(x)$, $\forall x \in D$.

Definição 3.2 (Função Composta): Sejam $g : D \rightarrow \mathbb{R}$ e $f : \mathbb{R} \rightarrow \mathbb{R}$ funções reais de variável real. A **função composta** $f \circ g : D \rightarrow \mathbb{R}$ é definida por

$$(f \circ g)(x) = f[g(x)]$$

Definição 3.3 (Função real de duas variáveis): Seja $X \subset \mathbb{R}^2$. A correspondência $f : X \rightarrow \mathbb{R}$ que associa a cada ponto $(x, y) \in X$ um único valor real $f(x, y)$ é chamada de função real de duas variáveis.

O conjunto X é o domínio de f e sua imagem é o conjunto de valores de f , ou seja, $\text{Im}(f) = \{f(x, y) \mid (x, y) \in X\}$.

Definição 3.4 (Limite de uma Função): Seja $f : X \rightarrow \mathbb{R}$ uma função, onde X contém pontos arbitrariamente próximos de $(a, b) \in \mathbb{R}^2$. Dizemos que o **limite** de $f(x, y)$ quando

(x, y) tende a (a, b) é L e escrevemos

$$\lim_{(x,y) \rightarrow (a,b)} f(x,y) = L$$

se para todo número $\epsilon > 0$ houver um número correspondente de $\delta > 0$ tal que, se $(x, y) \in X$ e $0 < \sqrt{(x-a)^2 + (y-b)^2} < \delta$, então $|f(x, y) - L| < \epsilon$.

Definição 3.5 (Função Contínua): Uma função $f : X \rightarrow \mathbb{R}$ é dita contínua em (a, b) se

$$\lim_{(x,y) \rightarrow (a,b)} f(x,y) = f(a,b)$$

Dizemos que f é **contínua** em X se f for contínua em todo ponto (a, b) de X .

Definição 3.6 (Derivadas Parciais): Se $f : X \rightarrow \mathbb{R}$ é uma função de duas variáveis, suas **derivadas parciais** são as funções f_x e f_y definidas por:

$$f_x(x,y) = \lim_{h \rightarrow 0} \frac{f(x+h,y) - f(x,y)}{h}, \quad f_y(x,y) = \lim_{h \rightarrow 0} \frac{f(x,y+h) - f(x,y)}{h}.$$

Observação 1. As derivadas parciais f_x e f_y indicam a taxa de variação de f em relação às variáveis x e y , respectivamente, mantendo-se a outra variável constante.

Definição 3.7 (Incremento): Considere uma função de duas variáveis, $z = f(x, y)$, e suponha que x varie de a para $a + \Delta x$ e y varie de b para $b + \Delta y$. Então, o **incremento** correspondente de z é

$$\Delta z = f(a + \Delta x, b + \Delta y) - f(a, b)$$

Definição 3.8 (Função Diferenciável): Se $z = f(x, y)$, então f é **diferenciável** em (a, b) se Δz puder ser expresso na forma

$$\Delta z = f_x(a,b)\Delta x + f_y(a,b)\Delta y + \epsilon_1\Delta x + \epsilon_2\Delta y$$

onde ϵ_1 e $\epsilon_2 \rightarrow 0$ quando $(\Delta x, \Delta y) \rightarrow (0, 0)$.

Definição 3.9 (Derivada Direcionada): A **derivada direcionada** de f em (x_0, y_0)

na direção do vetor unitário $u = (a, b)$ é

$$D_u f(x_0, y_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + ha, y_0 + hb) - f(x_0, y_0)}{h}$$

se esse limite existir.

Definição 3.10 (Função Vetorial): Seja $X \subset \mathbb{R}^2$. Uma função $\mathbf{F} : X \rightarrow \mathbb{R}^2$ que associa a cada ponto $x \in X$ um vetor:

$$\mathbf{F}(x) = (f(x), g(x))$$

onde $f, g : X \rightarrow \mathbb{R}$ são as funções componentes de \mathbf{F} , é chamada de função vetorial.

Definição 3.11 (Vetor Gradiente): Seja $f : X \rightarrow \mathbb{R}$ uma função real de duas variáveis. Então, o **gradiente** de f é a função vetorial $\nabla f : X \rightarrow \mathbb{R}^2$ definida por:

$$\nabla f(x, y) = (f_x(x, y), f_y(x, y))$$

Observação 2. O conceito de gradiente se generaliza para funções reais de n variáveis $f(x_1, \dots, x_n)$:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right),$$

o que é usado para otimização de funções de muitos parâmetros, como no caso do vetor de pesos α de uma RNA.

Teorema 3.12: Seja $f : X \rightarrow \mathbb{R}$ uma função diferenciável. O valor máximo da derivada direcional $D_u f(x)$ é $|\nabla f(x)|$ e ocorre quando u tem a mesma direção do vetor gradiente $\nabla f(x)$.

Observação 3. Para encontrar os valores mínimos de uma função f , usamos o negativo do vetor gradiente de f , ou seja, movemo-nos na direção oposta.

Definição 3.13 (Ponto de Fronteira): Seja $X \subset \mathbb{R}^2$ um conjunto. Um ponto $x \in \mathbb{R}^2$ é dito um **ponto de fronteira** de X se toda vizinhança de x contém pelo menos um ponto de S e pelo menos um ponto que não pertence a X . O conjunto de todos os pontos de fronteira de X é chamado **fronteira de X** , e é denotado por ∂X .

Definição 3.14 (Ponto Crítico): Um ponto $(a, b) \in X$ é chamado **ponto crítico** (ou ponto estacionário) de f se $f_x(a, b) = 0$ e $f_y(a, b) = 0$, ou se uma das derivadas parciais não existir.

Definição 3.15 (Mínimos e Máximos): Uma função de duas variáveis tem um **mínimo local** em (a, b) se $f(x, y) \geq f(a, b)$ quando (x, y) está próximo de (a, b) . O número $f(a, b)$ é chamado **valor mínimo local**. Analogamente, máximo local é definido.

Teorema 3.16: Se f tem um máximo ou mínimo local em (a, b) e as derivadas parciais de primeira ordem existem, então $f_x(a, b) = 0$ ou $f_y(a, b) = 0$.

Teorema 3.17 (Valor Extremo para Funções de Duas Variáveis): Se f é contínua em um conjunto fechado e limitado D em \mathbb{R}^2 , então f assume um valor máximo absoluto $f(x_1, y_1)$ e um valor mínimo absoluto $f(x_2, y_2)$ em alguns pontos (x_1, y_1) e (x_2, y_2) de D .

Observação 4. Funções de perda em RNAs (que serão apresentadas no capítulo seguinte) geralmente não são convexas, podendo apresentar múltiplos mínimos locais. O gradiente indica apenas a direção de descida, não garantindo mínimo global. Funções convexas são mais simples, pois todo ponto crítico é mínimo absoluto.

Definição 3.18 (Conjunto Convexo): Um conjunto $X \subset \mathbb{R}^2$ chama-se **convexo** quando o segmento de reta que une dois quaisquer de seus pontos está inteiramente contido em X , isto é,

$$x, y \in X, 0 \leq t \leq 1 \implies (1 - t)x + ty \in X$$

Definição 3.19 (Função Convexa): Seja $X \subset \mathbb{R}^2$ um conjunto convexo. Uma função $f : X \rightarrow \mathbb{R}$ chama-se **convexa** quando, para quaisquer $x, y \in X$ e $t \in [0, 1]$, tem-se

$$f((1 - t)x + ty) \leq (1 - t)f(x) + tf(y).$$

Teorema 3.20: Todo ponto crítico de uma função convexa é um ponto de mínimo absoluto.

Definição 3.21 (Função côncava): Uma função $f : X \rightarrow \mathbb{R}$ diz-se **côncava** quando a

função $-f$ é convexa, isto é,

$$f[(1-t)x + ty] \geq (1-t)f(x) + tf(y), \quad \forall x, y \in X, t \in [0,1].$$

Teorema 3.22: Todo ponto crítico de uma função côncava é um ponto de máximo absoluto.

3.2 Noções de Álgebra Linear

Além dos conceitos apresentados, para facilitar a compreensão de leitores de diferentes áreas, apresentamos a seguir alguns conceitos matemáticos básicos de álgebra linear. Estes conceitos serão utilizados, também, no capítulo seguinte.

Definição 3.23 (Matriz): Uma sequência (v_1, \dots, v_n) de vetores $v_j = (a_{1j}, \dots, a_{mj}) \in \mathbb{R}^m$ define uma **matriz A** de ordem $m \times n$ (m -linhas por n -colunas.), ou seja,

$$\mathbf{A} = [v_1, \dots, v_n] = [a_{ij}]_{m \times n}$$

Definição 3.24 (Transposta de Matriz): Seja a matriz $\mathbf{A} = [a_{ij}]_{m \times n}$. A **transposta** de \mathbf{A} , denotada \mathbf{A}^T , é definida por

$$\mathbf{A}^T = [a_{ji}]_{n \times m}.$$

Definição 3.25 (Soma de matrizes): Sejam $\mathbf{A} = [a_{ij}]$ e $\mathbf{B} = [b_{ij}]$ duas matrizes de ordem $m \times n$. A **soma das matrizes A e B** é a matriz $\mathbf{C} = [c_{ij}]$ obtida pela soma termo a termo dos elementos correspondentes, isto é:

$$c_{ij} = a_{ij} + b_{ij}, \quad \text{para } i = 1, 2, \dots, m \text{ e } j = 1, 2, \dots, n.$$

Assim, a operação de soma entre matrizes só é possível quando ambas possuem a mesma ordem.

Definição 3.26 (Produto de matrizes): Seja \mathbf{A} uma matriz de ordem $m \times n$ e \mathbf{B} uma matriz de ordem $n \times p$. O **produto de matrizes A e B** é definido como a matriz

$\mathbf{C} = [c_{ij}]$ de ordem $m \times p$, em que cada elemento c_{ij} é dado por:

$$c_{ij} = \sum_{k=1}^n a_{ik}b_{kj}, \quad \text{para } i = 1, 2, \dots, m \text{ e } j = 1, 2, \dots, p.$$

A multiplicação entre matrizes só é possível quando o número de colunas de \mathbf{A} é igual ao número de linhas de \mathbf{B} .

Observação 5. As definições 3.25 e 3.26 são muito utilizadas no contexto das RNAs pois em cada camada é feito o produto entre a matriz de pesos e o vetor de entradas e, em seguida, é adicionado o vetor de *bias*.

Definição 3.27 (Transformação Linear): Uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ é chamada **transformação linear** se, para quaisquer $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ e $c \in \mathbb{R}$, temos:

$$f(\mathbf{u} + \mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v})$$

$$f(c\mathbf{u}) = cf(\mathbf{u}).$$

Definição 3.28 (Função Afim): Uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é **afim** se pode ser escrita como $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$, onde $\mathbf{w} \in \mathbb{R}^n$ e $b \in \mathbb{R}$, onde “ \cdot ” representa o produto escalar.

Definição 3.29 (Norma Euclidiana de Vetor): A norma euclidiana de um vetor $\mathbf{v} = (v_1, \dots, v_n) \in \mathbb{R}^n$ é

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2}.$$

Os conceitos apresentados neste capítulo fornecem a base matemática para compreender os mecanismos de aprendizado das RNAs. A partir dessas ideias, especialmente da noção de gradiente, derivadas, normas, produtos internos e minimização de funções, é possível entender como os modelos ajustam seus parâmetros durante o treinamento. No próximo capítulo, exploraremos o funcionamento teórico das RNAs, detalhando suas estruturas, funções de ativação e processos de treinamento.

4 Redes Neurais Artificiais: estudo teórico

O cérebro humano funciona de maneira interessante, pois é capaz de adaptar seus comportamentos e usar suas experiências para obter aprendizado. Inspiradas nesse funcionamento, as RNAs¹ são modelos computacionais e matemáticos com capacidade de aprendizado, permitindo a tomada de decisões com base no que se aprendeu.

Segundo Baroni e Padilha (2021), o sucesso das RNAs se deve à sua capacidade de resolver diversos problemas complexos, nas mais variadas situações da vida real. De acordo com Furtado (2019), graças ao trabalho incessante de diversos pesquisadores, os fundamentos teóricos estão se tornando cada vez mais sólidos, e novas aplicações têm surgido em áreas como engenharia, economia, agronomia e medicina. Entre os problemas abordados destacam-se a extração de características, classificação, categorização/clusterização, estimativa e previsão, otimização e aproximação de funções.

Nesse contexto, compreender a estrutura e o funcionamento das RNAs torna-se fundamental para analisar seu potencial e suas limitações. Segundo Furtado (2019), as RNAs podem ser vistas como uma tentativa de simular, ainda que de forma simplificada, os mecanismos de aprendizado presentes no cérebro humano. Elas são compostas por unidades artificiais interconectadas, capazes de processar informações em paralelo e de ajustar seus parâmetros internos a partir da experiência. Esse processo de ajuste permite que a rede aprenda padrões complexos a partir dos dados, o que explica sua ampla aplicação em problemas que envolvem reconhecimento de padrões, análise preditiva e tomada de decisão.

As RNAs podem ser organizadas de diversas formas, de acordo com a maneira como suas camadas e conexões são estruturadas. As do tipo *feedforward* são caracterizadas pelo fluxo unidirecional de informações, em que os sinais percorrem as camadas de entrada,

¹Neste trabalho, a sigla RNA será usada para se referir a uma rede específica, enquanto RNAs será usada no plural, para se referir de forma geral.

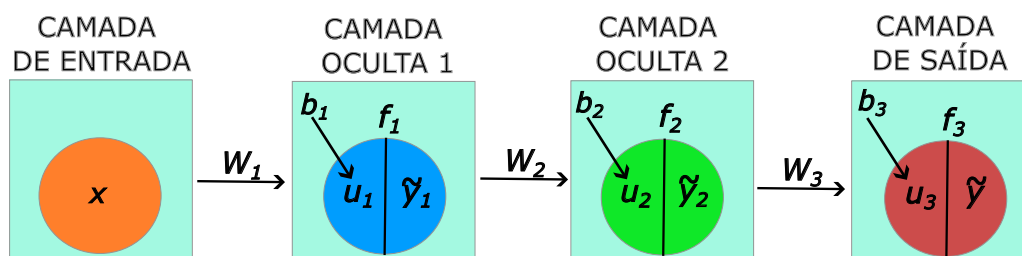
intermediárias e de saída sem que ocorram realimentações (CARVALHO, 2024). Essa arquitetura apresenta uma organização semelhante à do córtex humano, com neurônios dispostos em camadas paralelas e consecutivas, propagando as informações sempre no mesmo sentido (FURTADO, 2019). Por essa razão, redes *feedforward* são amplamente utilizadas em aplicações supervisionadas, como em problemas de classificação e regressão.

Entre as arquiteturas mais conhecidas desse tipo, temos uma das primeiras e mais simples formas de RNAs: o *Perceptron*, idealizado por Frank Rosenblatt (Estados Unidos) na década de 1950, que possuía uma única camada. Segundo Carvalho (2024), embora simples, o *Perceptron* abriu caminho para avanços significativos nas décadas seguintes, aperfeiçoando, a partir da introdução do algoritmo de retropropagação do erro (*backpropagation*) em 1980, o desenvolvimento de algo mais sofisticado: a RNA *Perceptron* Multicamadas (MLP). Enquanto uma rede *Perceptron* é capaz de resolver problemas básicos de classificação binária, uma rede neural com múltiplas camadas pode lidar com problemas mais complexos, como de classificação multiclasse, reconhecimento de imagens, tradução automática de idiomas, entre outras aplicações (CARVALHO, 2024).

De acordo com Manzan (2016), a MLP é um dos modelos mais difundidos. Nela, os neurônios estão organizados em diferentes camadas, cada uma com uma função no processamento da informação: camada de entrada, camadas ocultas e camada de saída. Além disso, as redes MLPs serviram como base conceitual para o desenvolvimento de outras classes de redes neurais, como as Convolucionais (*Convolutional Neural Networks*) e as Recorrentes (*Recurrent Neural Networks*), cada uma delas projetada para lidar com tipos específicos de dados e desafios computacionais (CARVALHO, 2024).

Neste trabalho, utilizaremos a rede MLP como principal arquitetura de rede neural. A Figura 4.1 apresenta um exemplo dessa arquitetura.

Figura 4.1: RNA *Perceptron* de quatro camadas



Fonte: Elaboração dos autores (2025).

Sendo:

- x : Sinal de entrada (correspondente ao atributo)
- y : Saídas reais
- W : Pesos
- b : *Bias*
- f_1, f_2 e f_3 : Funções de Ativação

Além disso, as operações matemáticas correspondentes são:

- $u_1 = \sum W_1 x + b_1$
- $\tilde{y}_1 = f_1(u_1)$: saída da camada oculta 1
- $u_2 = \sum W_2 \tilde{y}_1 + b_2$
- $\tilde{y}_2 = f_2(u_2)$: saída da camada oculta 2
- $u_3 = \sum W_3 \tilde{y}_2 + b_3$
- $\tilde{y} = f_3(u_3)$: saída do modelo RNA

As operações matemáticas correspondentes a cada camada foram descritas por expressões escalares, que serão posteriormente generalizadas na forma vetorial no Capítulo 6.

Nesta simples RNA de quatro camadas é possível observar que:

- Há apenas um atributo;
- Cada camada possui apenas um neurônio;
- Há uma função de ativação em cada camada oculta e também na camada de saída.

Vale salientar que essas características referem-se especificamente ao exemplo da Figura 4.1, podendo variar de forma significativa em outras RNAs. Dentre os vários elementos que podem diferenciá-las, há componentes, parâmetros e hiperparâmetros.

Tabela 4.1: Classificação dos elementos de uma Rede Neural Artificial

Categoria	Exemplos
Componentes	<ul style="list-style-type: none">• Camada de entrada;• Camadas ocultas;• Camada de saída;• Neurônios;• Atributos.
Parâmetros	<ul style="list-style-type: none">• Pesos;• <i>Bias</i>.
Hiperparâmetros	<ul style="list-style-type: none">• Estruturais: número de camadas ocultas, número de neurônios por camada, função de ativação;• De treinamento: taxa de aprendizagem, número de épocas, tamanho do lote (<i>batch size</i>), otimizador, <i>momentum</i>;• Complementar: Função de custo.

Fonte: Elaboração dos autores (2025).

A fim de facilitar a compreensão, a Tabela 4.1 apresenta uma classificação resumida dos elementos que constituem as RNAs, os quais serão detalhados nas seções seguintes.

Embora este capítulo apresente o estudo teórico das RNAs em nível conceitual, nos capítulos 6 e 7 aboremos suas aplicações práticas.

4.1 Componentes de uma RNA

Os componentes das RNAs mencionados acima são essenciais para o funcionamento da rede. A seguir, detalhamos sobre cada um deles.

4.1.1 Camada de Entrada (*Input*)

É a primeira camada da RNA, responsável por receber os dados a serem processados. Nessa camada, cada neurônio representa uma variável da amostra (ou característica do dado de entrada). Os valores recebidos são repassados para a próxima camada inalterados organizados em forma de vetor.

Observação 6. O vetor de saída de uma camada é utilizado como vetor de entrada para a camada seguinte.

4.1.2 Camadas Ocultas (*Hidden Layers*)

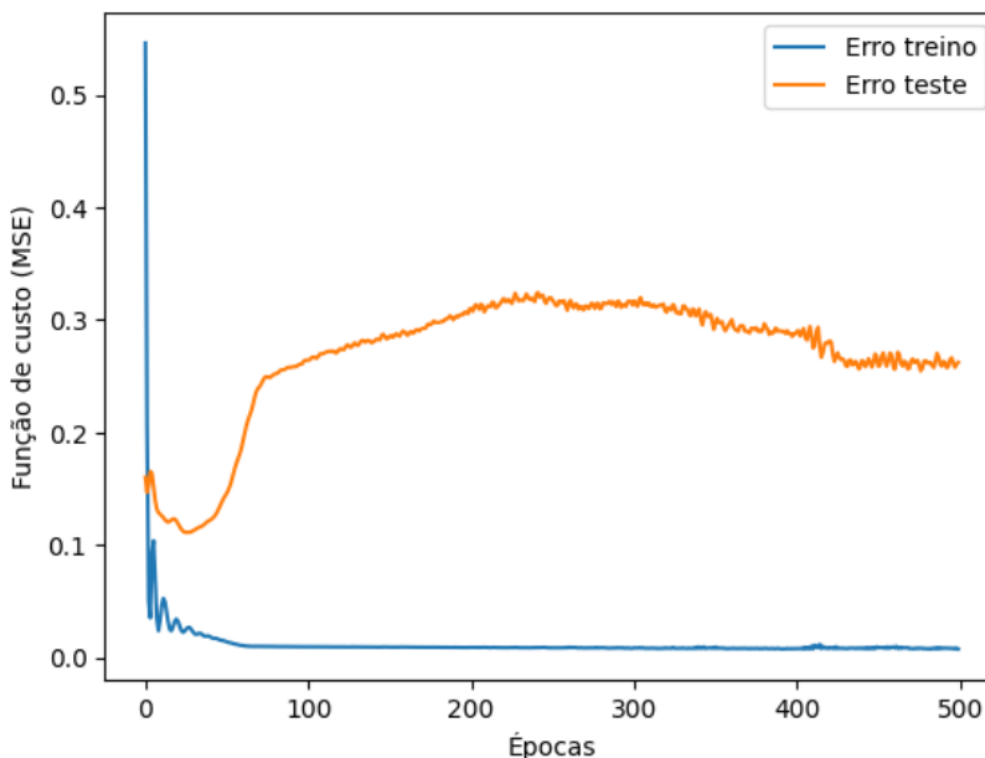
Segundo Furtado (2019), a função das camadas ocultas é processar e transformar os dados provenientes da camada de entrada antes de encaminhá-los à camada de saída. Cada camada recebe o vetor com dados de saída da camada anterior, extraindo padrões e aplicando ajustes com os pesos. Em seguida, esses valores são somados e é aplicada uma função de ativação (se houver). Após esse processo, os dados atualizados são repassados para a próxima camada também organizados em forma de vetor. Esse processo se repete em todas as camadas ocultas até que, por fim, os dados cheguem à camada de saída.

O número de neurônios nas camadas ocultas não é fixo. Brasil *et al.* classificam RNAs como simples quando possuem uma ou poucas camadas ocultas e como redes profundas quando apresentam muitas. Caso a rede consiga aprender, pode-se tentar reduzir o número de neurônios; caso contrário, recomenda-se aumentá-los (FURTADO, 2019). Não há quantidade pré-definida de camadas ocultas, e a adição de uma ou mais delas pode melhorar o mapeamento de problemas mais complexos. Entretanto, segundo Furtado (2019), no máximo duas camadas ocultas, com número suficiente de neurônios por camada, são suficientes para produzir quaisquer mapeamentos. Complementarmente, Silva, Spatti e Flauzino (2016) afirmam que RNAs com apenas uma camada oculta tendem a ser menos propensas a ficarem presas em mínimos locais, pois sua estrutura mais compacta reduz a complexidade geométrica da função que mapeia o erro quadrático médio.

Portanto, vale ressaltar que o aumento desordenado de neurônios e/ou de camadas ocultas não garantem a generalização adequada da RNA em relação às amostras pertencentes aos subconjuntos de teste. O excesso pode causar, inclusive, a circunstância de memorização excessiva (*overfitting*), em que a rede memoriza as respostas aos estímulos apresentados nas entradas (SILVA, SPATTI e FLAUZINO, 2016). Nesses casos, o erro

quadrático da amostragem de treino tende a ser bem baixo. Entretanto, o erro quadrático da amostragem de teste pode assumir valores bem elevados, fato que denota a condição de *overfitting*. Essa situação de *overfitting* é ilustrada na Figura 4.2.

Figura 4.2: Representação de um *overfitting*.



Fonte: Elaboração dos autores (2025).

4.1.3 Camada de Saída (*Output*)

A camada de saída é a última camada da RNA, responsável por gerar uma resposta final. Essa resposta pode ser um valor contínuo (caso seja um problema de regressão) ou um valor que represente uma classe ou uma probabilidade (caso seja um problema de classificação). Além disso, essa camada pode conter múltiplos neurônios, cada um correspondendo a uma saída, que equivale a uma resposta do modelo.

Para compreender plenamente o funcionamento de uma RNA, é importante conhecer também seus outros componentes fundamentais, incluindo:

4.1.4 Neurônios

Neurônios (ou nós) são unidades de processamento de uma RNA. Em cada neurônio, calcula-se o sinal de entrada — que corresponde ao valor de saída do(s) neurônio(s) da camada anterior — multiplicado pelo peso, somado ao *bias*. Em seguida, aplica-se a função

de ativação, se houver. Embora aqui estejamos descrevendo o cálculo de cada neurônio de forma escalar, na prática ele ocorre de forma vetorial, ou seja, todos os neurônios de uma camada são calculados simultaneamente. Matematicamente, Carvalho (2024) define neurônios artificiais como:

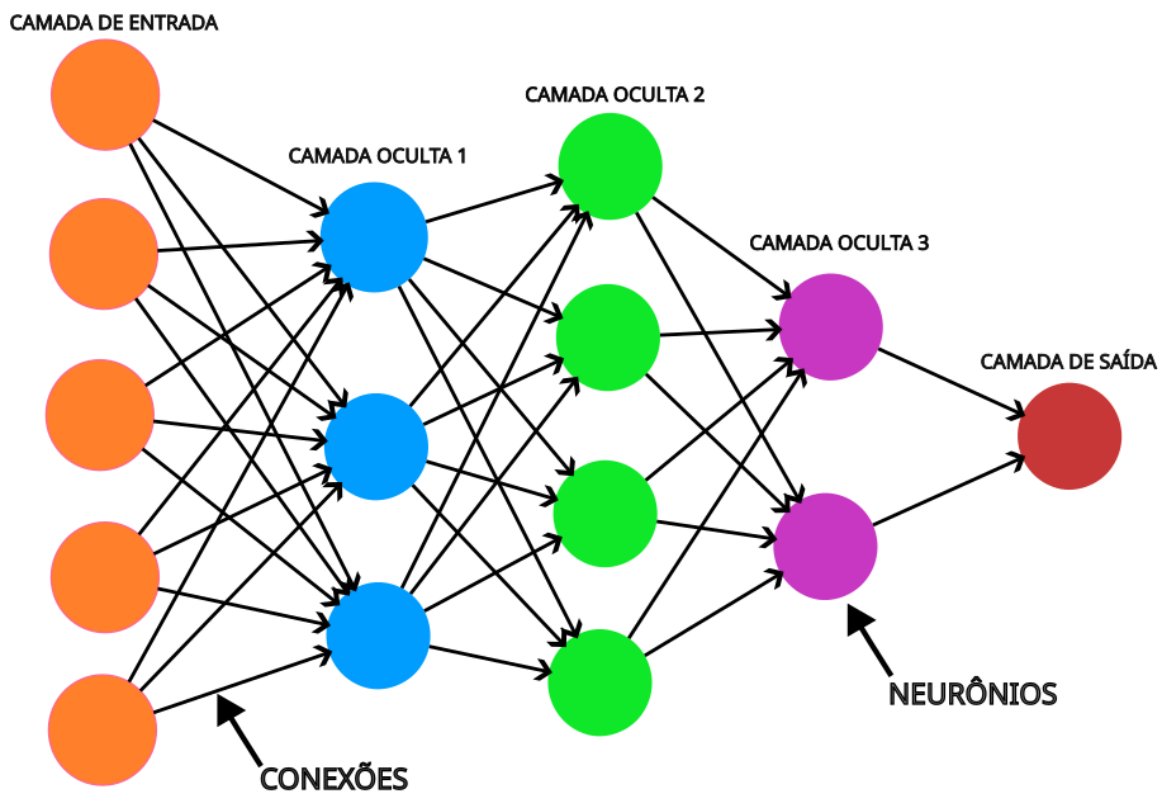
“Um Neurônio Artificial é uma unidade de processamento computacional que processa as informações de um conjunto $\vartheta = \{x_1, x_2, \dots, x_n\}$, chamado de entradas, de tal forma que dada uma função $f : \vartheta \subset R \rightarrow R$ o objeto de saída é dado da seguinte maneira

$$\tilde{y} = f\left(\sum_{i=1}^n x_i w_i + b_i\right),$$

em que os w_i 's são os pesos e b_i são chamados de viés (*bias*), provendo um ajuste linear seguido de uma aplicação de uma função.”

A Figura 4.3 apresenta a representação de uma RNA com seus neurônios e conexões. Nesta RNA, é possível observar que cada neurônio está conectado a neurônios da camada seguinte, o que é caracteriza a rede como densa. Além disso, nessa figura é ilustrando o fluxo de informações na RNA.

Figura 4.3: Neurônios e conexões de uma RNA.



Fonte: Elaboração dos autores (2025).

4.1.5 Atributos

Os atributos são variáveis qualitativas ou quantitativas que descrevem o problema em estudo. A quantidade de atributos determina o número de neurônios na primeira camada, isto é, a quantidade de componentes do vetor de saída da camada de entrada.

Como afirmam Finger (2021), os atributos podem assumir diferentes formatos, como valores numéricos, textos ou imagens. Por isso, pode ser necessário pré-processar os dados, através de normalizações ou conversões de formato para torná-los mais apropriados à sua utilização na rede (FURTADO, 2019).

4.2 Parâmetros

Os parâmetros correspondem aos valores ajustados pela rede durante o processo de treinamento, sendo atualizados a cada época.

4.2.1 Pesos

Os pesos (W) são valores associados em cada conexão entre neurônios. Inicialmente, os pesos são atribuídos de forma aleatória, mas durante o treinamento da rede (juntamente com os *bias*) eles são ajustados para minimizar o erro da rede. Os pesos de cada camada podem ser organizados em uma matriz de pesos, facilitando o cálculo vetorial das ativações. Segundo Baroni e Padilha (2021), o ajuste dos pesos constitui o principal mecanismo de treinamento de uma RNA.

4.2.2 Bias

O *bias* (ou viés) é um valor adicionado à soma ponderada das entradas de um neurônio. De forma escalar, o sinal de entrada do neurônio é calculado como $Wx + b$, onde x é a entrada, W o peso e b o *bias*. De forma vetorial, o *bias* é representado como um vetor somado ao produto entre a matriz de pesos e o vetor de saída da camada anterior. Esta parte é linear no modelo, em que o peso controla a inclinação da reta e o *bias* permite mover a reta para cima ou para baixo. Diferente dos pesos, os *bias* não estão presentes nas conexões, mas nos neurônios. Segundo Furtado (2019), o *bias* mantém fidelidade ao modelo biológico e flexibilidade ao modelo computacional.

Assim como os pesos, o *bias* é aprendido e ajustado durante o treinamento da rede. Vale salientar que ambos são de extrema importância em uma RNA, pois permitem que o

modelo ajuste a saída dos neurônios e aprenda funções mais complexas.

4.3 Hiperparâmetros

Os hiperparâmetros são valores definidos pelo projetista da rede antes do início do treinamento. Diferentemente dos parâmetros da rede, eles não são aprendidos automaticamente e devem ser ajustados manualmente ou por meio de técnicas de busca de hiperparâmetros.

A escolha adequada de hiperparâmetros, como taxa de aprendizagem e estrutura da rede, é um fator crítico para o desempenho da RNA. Conforme destaca Manzan (2016), essas definições influenciam diretamente a capacidade de generalização do modelo, podendo comprometer o aprendizado se forem mal configuradas.

Embora o número de camadas ocultas e o número de neurônios por camada já tenham sido abordados na seção anterior sobre componentes da rede, é importante destacar que tais quantidades também configuram hiperparâmetros estruturais.

4.3.1 Função de Ativação

Após a combinação linear entre os pesos e o *bias*, uma função de ativação é aplicada ao resultado dessa soma para determinar a saída do neurônio. Algebricamente:

$$\tilde{y} = f\left(\sum Wx + b\right),$$

onde \tilde{y} é o valor de saída do neurônio, f é a função de ativação, W representa a matriz de pesos, x é o vetor de entrada do neurônio (ou saída da camada anterior) e b é o vetor de *bias*.

A função de ativação introduz não-linearidade no modelo, permitindo que a RNA aprenda padrões complexos. Ela é aplicada dentro dos neurônios, nas camadas ocultas e, quando necessário, também na camada de saída. Na última camada, a função de ativação usada pode também ser chamada de função de saída. Ela é usada para produzir o resultado final com base nos cálculos das camadas anteriores e é escolhida de acordo com o tipo de problema que a rede precisa resolver (classificação, regressão, etc).

Além disso, a função de ativação limita o valor de saída do neurônio, que é então transmitido para a próxima camada da rede neural. Segundo Furtado (2019), existem duas condições tradicionalmente associadas ao uso de uma função matemática como função de

ativação em uma RNA:

1. a função deve ser contínua;
2. os limites da função quando $x \rightarrow \infty$ devem ser finitos.

Embora Furtado destaque essas condições, elas não são estritamente necessárias em todos os contextos. Tais requisitos devem ser compreendidos como condições clássicas e suficientes em determinados modelos teóricos, mas não como restrições universais. Funções amplamente utilizadas em redes neurais modernas, como a ReLU, não possuem limite superior finito, uma vez que $\lim_{x \rightarrow +\infty} \text{ReLU}(x) = +\infty$. Ainda assim, isso não compromete seu uso prático, pois eventuais problemas de instabilidade são controlados por técnicas de treinamento.

Vale destacar que, em redes neurais, a função de ativação pode ser aplicada **elemento a elemento** em um vetor de entradas $\mathbf{x} = [x_1, x_2, \dots, x_n]^\top$, ou seja:

$$\mathbf{f}(\mathbf{x}) = [f(x_1), f(x_2), \dots, f(x_n)]^\top,$$

exceto em casos como a função *Softmax*, que depende de todo o vetor de saída.

Entre as funções de ativação mais utilizadas, destacam-se:

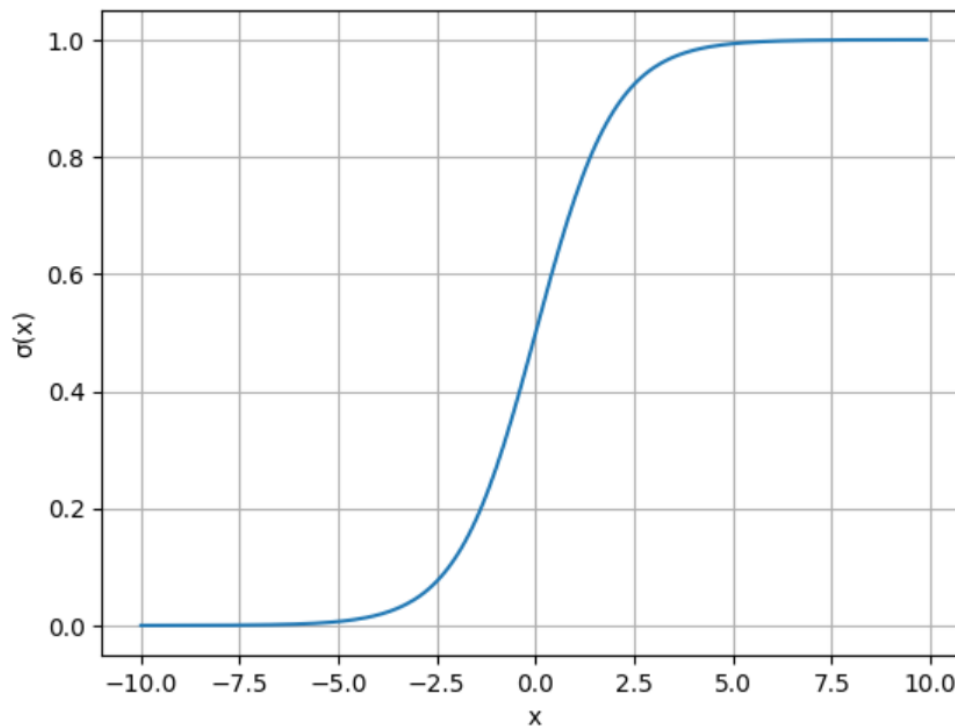
Função Sigmoid

A Função Sigmoid (ou Função Logística) é uma das funções de ativação mais conhecidas e utilizadas em RNAs. Ela transforma qualquer valor real de entrada em um valor entre 0 e 1, o que é útil para problemas de classificação binária. É definida como:

$$\sigma(x) = \frac{1}{1 + e^{-x}},$$

onde x representa a entrada do neurônio. Nelson (2023) afirma que a saída da Função Sigmoid pode ser interpretada como a probabilidade de um dado pertencer a uma determinada classe. O comportamento da Função Sigmoid é ilustrado na Figura 4.4.

Figura 4.4: Comportamento da Função Sigmoide



Fonte: Elaboração dos autores (2025).

Apesar de sua ampla utilização, Rizzo e Canato (2020) destacam que a Função Sigmoide pode saturar para valores muito altos ou baixos, fazendo com que o gradiente se aproxime de zero e prejudique o aprendizado (desvanecimento do gradiente).

Função ReLU (Rectified Linear Unit)

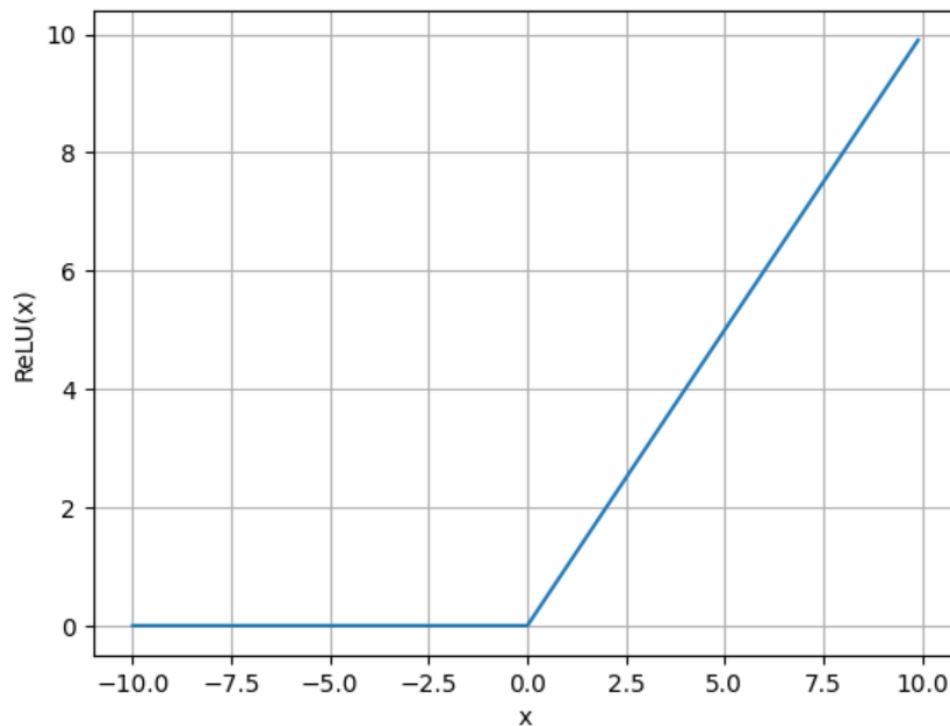
A Função ReLU é popular por sua simplicidade e eficiência computacional. Ela retorna apenas valores não negativos, sendo definida como:

$$f(x) = \max(0, x), \quad f(x) \text{ denominada ReLU,}$$

onde x é a entrada do neurônio. O comportamento da Função ReLU é ilustrado na Figura 4.5.

Segundo Rizzo e Canato (2020), a ReLU apresenta menor risco de desvanecimento do gradiente em valores positivos, mas pode sofrer do problema do *dying ReLU* para entradas negativas.

Figura 4.5: Comportamento da Função ReLU



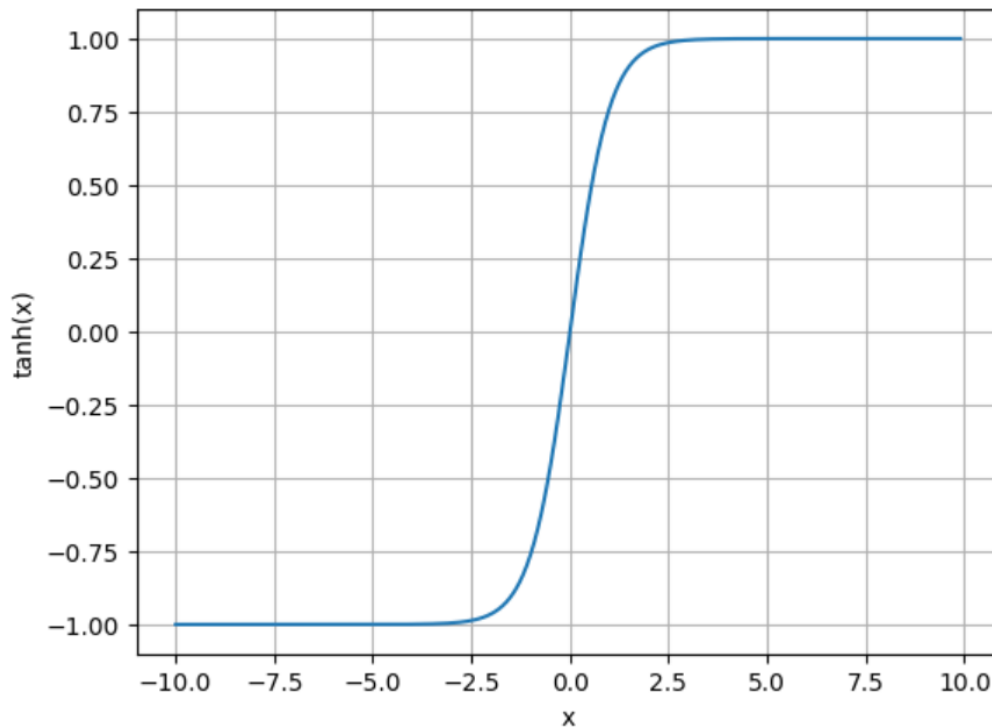
Fonte: Elaboração dos autores (2025).

Função Tangente Hiperbólica

A Função Tangente Hiperbólica (Tanh) é suave e contínua, com saída entre -1 e 1, o que ajuda a centralizar os dados em zero. É definida como:

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1},$$

onde x é a entrada do neurônio. O comportamento da Tanh é mostrado na Figura 4.6.

Figura 4.6: Comportamento da Função Tangente Hiperbólica

Fonte: Elaboração dos autores (2025).

Silva, Spatti e Flauzino (2016) destacam que a Tangente Hiperbólica, por ser uma função ímpar, ajuda na convergência durante o treinamento. Porém, também pode sofrer desvanecimento do gradiente em entradas extremas.

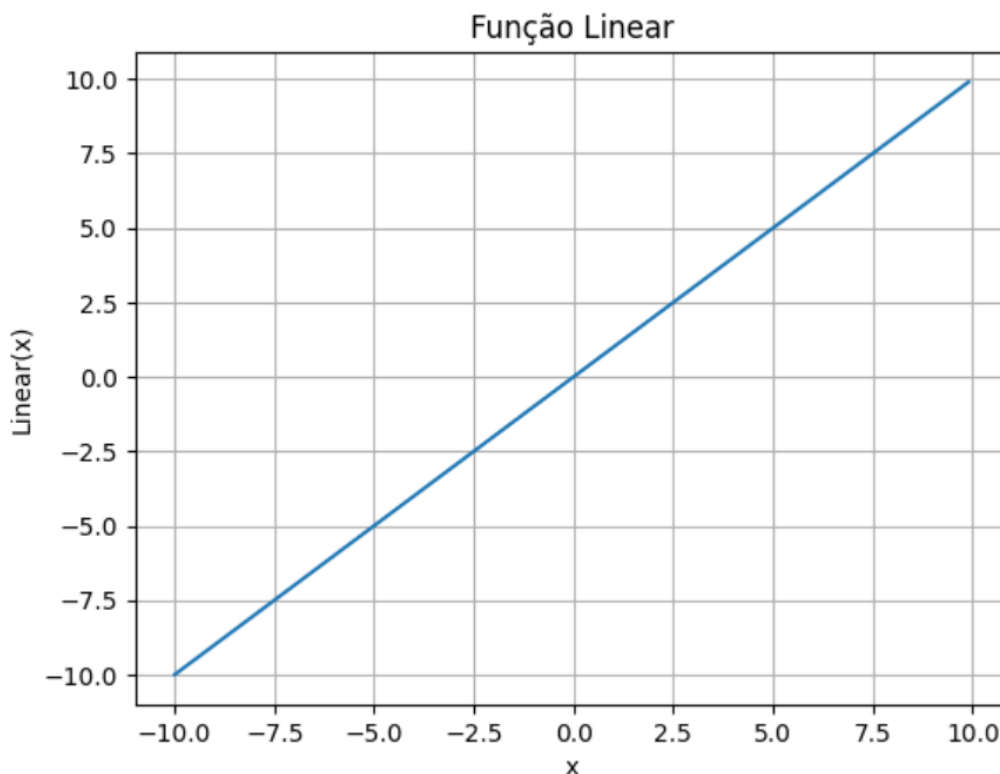
Função Linear

A Função Linear (ou Identidade) é a mais simples possível, definida como:

$$f(x) = x,$$

onde x é a entrada do neurônio. Apesar de ser função de ativação, não introduz não-linearidade. O comportamento da Função Linear é ilustrado na Figura 4.7.

Figura 4.7: Comportamento da Função Linear



Fonte: Elaboração dos autores (2025).

Furtado (2019) recomenda usar Função Linear na última camada quando a saída for contínua, sendo prática para problemas de regressão (Rizzo e Canato, 2020).

Função *Softmax*

A função *Softmax* é amplamente utilizada na camada de saída de RNAs aplicadas a problemas de classificação multiclasse. Sua principal função é transformar um vetor de entradas reais $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ em um vetor de probabilidades associado a cada classe possível do problema, conforme a expressão:

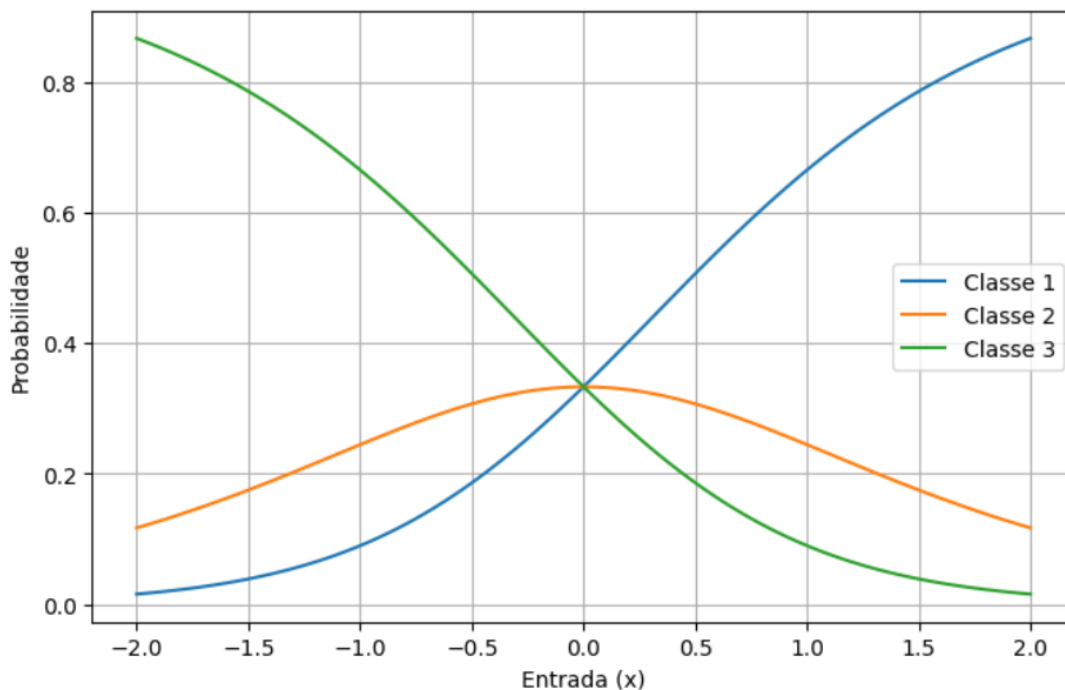
$$\mathbf{f}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}, \quad i = 1, \dots, n.$$

Cada componente do vetor de saída corresponde à probabilidade estimada pelo modelo de a entrada pertencer a uma determinada classe, sendo garantido que a soma dessas probabilidades seja igual a 1.

Um exemplo do comportamento da Função *Softmax* é ilustrado na Figura 4.8. Diferentemente dos gráficos das funções anteriores, a função *Softmax* representa mais de uma curva. Nesse contexto, as curvas denominadas Classe 1, Classe 2 e Classe 3

representam as probabilidades associadas a cada uma das possíveis categorias de saída do modelo, sendo cada classe vinculada a um neurônio da camada de saída da rede neural.

Figura 4.8: Comportamento da Função *Softmax*



Fonte: Elaboração dos autores (2025).

Observa-se que, para cada valor da entrada x , a função *Softmax* distribui a probabilidade entre as classes, atribuindo maior valor àquela considerada mais provável pelo modelo. À medida que a entrada varia, a probabilidade de uma classe pode aumentar, enquanto as demais diminuem, mantendo-se sempre a soma total igual a 1. Assim, a decisão do modelo é dada pela classe que apresenta a maior probabilidade para uma determinada entrada.

Em aplicações práticas, como na triagem inicial do TEA, essas classes podem representar diferentes categorias de risco ou perfis de resposta, permitindo que o modelo indique qual classe é mais compatível com os dados fornecidos. Segundo Carvalho (2024), a função *Softmax* também favorece o cálculo do gradiente durante o treinamento da rede, por ser composta por funções infinitamente diferenciáveis, o que contribui para a eficiência do processo de aprendizado.

A escolha da função de ativação depende do tipo de problema, da velocidade e eficiência do treinamento, da arquitetura da RNA, do custo computacional e do tipo de saída desejada (Brasil *et al.*). A Tabela 4.2 apresenta as aplicações típicas de cada função.

Tabela 4.2: Principais funções de ativação e seus tipos de aplicação

Função	Aplicação típica
Sigmoide	Classificação binária
ReLU	Redes profundas ou camadas ocultas de diferentes tipos de problemas
Tangente Hiperbólica	Diferentes tipos de problemas em camadas ocultas
Linear	Problemas de regressão
<i>Softmax</i>	Classificação multiclasse

Fonte: Elaboração dos autores (2025).

4.3.2 Função de Custo (*Loss*)

Segundo Carvalho (2024), a Função de Custo é uma medida de quão bem a rede está performando. Ela é aplicada aos dados de treinamento e quantifica o erro cometido pelo modelo ao realizar suas previsões, isto é, mede o erro entre a saída encontrada e a saída esperada ($\tilde{y} - y$) (NELSON, 2023). O objetivo do treinamento é minimizar essa função, ajustando os pesos corretamente. Mas não é necessário testar cada valor possível para o peso até encontrar o que resulta em um menor erro. E, mesmo se isso fosse feito, não seria possível determinar quando parar, visto que, não seria possível prever se existiriam valores melhores ao continuar. Por isso, é necessário conhecer a função de custo, suas propriedades matemáticas e aproveitar suas vantagens (NELSON, 2023).

Segundo Nelson (2023), para a escolha de uma Função de Custo, tendo em vista que buscamos sua minimização, as melhores funções para lidar são ou convexas (onde não há mínimos locais, uma vez que qualquer mínimo local também é um mínimo global) ou lineares (onde são ao mesmo tempo convexas e côncavas). Ao minimizar uma função não convexa, o processo pode levar ao encontro de mínimos locais, sem garantir que o mínimo global seja encontrado.

É incontestável que os minimizadores e maximizadores ocorrem em pontos críticos ou em pontos de fronteira. Portanto, para encontrar tais otimizadores, é necessário analisar tanto os pontos críticos internos quanto os pontos de fronteira (NELSON, 2023). De acordo com Nelson (2023), em geral, há duas abordagens para localizar minimizadores (e/ou maximizadores) de funções:

Abordagem 1: igualar a derivada da função de perda a zero, e então resolver para os

pesos.

Abordagem 2: seguir a direção do gradiente para descer em direção ao mínimo ou subir em direção ao máximo. O gradiente é o vetor das derivadas parciais de uma função de várias variáveis e indica a direção de maior crescimento da função. Na prática, como afirma Carvalho (2024), utilizamos o negativo do gradiente para mover na direção oposta a ele, ou seja, minimizar a função.

Algumas funções de custo serão apresentadas a seguir:

Função *Mean Square Error*

A Função Erro Quadrático Médio (Mean Square Error ou MSE) é definida por:

$$\text{MSE} = \frac{1}{m n_y} \sum_{i=1}^m \sum_{j=1}^{n_y} (\tilde{y}_j^{(i)} - y_j^{(i)})^2$$

onde m é o número total de amostras no conjunto de dados, n_y é o número de variáveis de saída, $y_j^{(i)}$ é o valor real da amostra i do j -ésimo vetor de saída, $\tilde{y}_j^{(i)}$ é o valor previsto (estimado pelo modelo) para a amostra i na componente j do vetor de saída.

Por ser quadrática em relação às previsões, sua derivada é linear, o que facilita a minimização pela Abordagem 1 e, conseqüentemente, a determinação dos pesos (NELSON, 2023).

Função *Cross-Entropy*

A Função Entropia Cruzada (*Cross-Entropy*) é uma das funções de perda mais utilizadas em problemas de classificação, especialmente quando a saída da rede representa probabilidades. Essa função mede a divergência entre a distribuição real das classes e a distribuição prevista pelo modelo. É definida, de forma geral, por:

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \cdot \log(p(y_{i,c}))$$

onde N é o número de amostras, C o número de classes, $y_{i,c}$ representa o valor real (geralmente 0 ou 1) da amostra i para a classe c , e $p(y_{i,c})$ é a probabilidade prevista para essa mesma classe.

Essa função penaliza fortemente previsões com alta confiança em classes incorretas, sendo, portanto, uma métrica eficaz para problemas em que o objetivo é maximizar a probabilidade da classe correta. Por ser uma função convexa em relação às probabilidades

previstas, ela favorece o processo de minimização utilizando a Abordagem 2, permitindo que o processo de otimização convirja para um ótimo global (NELSON, 2023).

Existem variações da Função Entropia Cruzada, que se aplicam conforme o tipo de problema de classificação:

- **Binary Cross-Entropy:** utilizada em problemas de classificação binária (duas classes). Sua forma específica é dada por:

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N \left[y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \right]$$

onde $y_i \in \{0,1\}$ indica a classe da amostra i , sendo 1 a classe positiva e 0 a classe negativa, $p(y_i)$ é a probabilidade prevista para a classe positiva e N é o número de amostras.

- **Categorical Cross-Entropy:** aplicada em problemas de classificação multiclasse, onde o modelo deve atribuir cada amostra a uma entre várias categorias possíveis. Nesse caso, a função soma as perdas em todas as classes.

Assim, a Função Cross-Entropy, em suas diferentes variações, desempenha papel essencial no treinamento de RNAs, guiando o ajuste dos pesos de modo a reduzir a discrepância entre as previsões do modelo e os valores reais.

Vale salientar que, embora a Função MSE permita, em casos simples como a regressão linear, a obtenção de solução analítica pela Abordagem 1, em modelos mais complexos ela também é minimizada por meio da Abordagem 2. Já a Função Cross-Entropy, em suas variações (*binary* e *categorical*), é naturalmente tratada com a Abordagem 2, beneficiando-se de sua convexidade para garantir convergência ao mínimo global durante o processo de treinamento.

Além disso, é importante destacar que a função de custo, embora seja uma escolha de modelagem, é tratada na prática como um hiperparâmetro, pois deve ser definida antes do início do treinamento.

4.3.3 Épocas (*epochs*)

Época corresponde a quantidade de vezes que a RNA vai percorrer todo o conjunto de dados para aprender, desde a camada de entrada até a camada de saída. Na primeira época, os pesos e *bias* são aleatórios, e são feitos os cálculos com cada amostra de dados.

Ao final desta época, é calculado o erro. Para a época seguinte, são selecionados novos valores para os pesos e *bias*, e todas as amostras de entrada são novamente processadas.

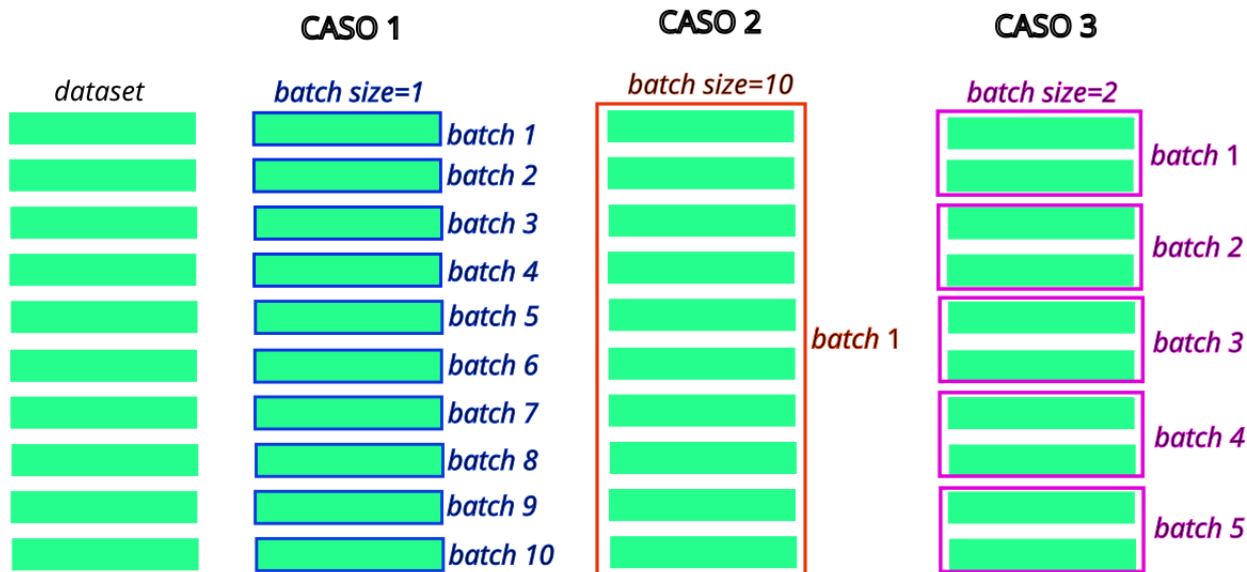
A quantidade de épocas dependerá do problema, e pode ser definida pelo projetista da rede ou até que o modelo esteja com erro mínimo e capacidade de generalização máxima. Um número reduzido de iterações pode impedir que o modelo alcance a convergência, enquanto um número excessivo pode resultar em desperdício de recursos computacionais e em sobreajuste, situação em que o modelo se adapta demais aos dados de treinamento, perdendo capacidade de generalização para novos dados (CARVALHO, 2024). Assim, além de definir a quantidade de épocas, muitas vezes são utilizados também critérios de parada para evitar o desperdício computacional e o tempo de processamento.

Segundo Silva, Spatti e Flauzino (2016), estabelecer uma quantidade máxima de épocas como critério de parada do algoritmo de treinamento, é uma estratégia simples e eficiente quando a precisão desejada se torna inalcançável.

Outro critério de parada bastante utilizado é a parada antecipada (*early stopping*). Neste método, o desempenho do modelo é monitorado constantemente nos subconjuntos de teste. O treinamento é interrompido quando a função de perda no conjunto de validação deixa de apresentar melhora e passa a aumentar em épocas sucessivas. Para controlar isso, define-se um parâmetro chamado *patience*, que determina o número máximo de épocas consecutivas sem melhora antes da interrupção do treinamento (SILVA, SPATTI E FLAUZINO, 2016).

4.3.4 *Batch Size*

Um aspecto relevante no treinamento de RNAs é a definição do *batch size* (tamanho do lote), isto é, o número de amostras processadas antes da atualização dos pesos da rede. Segundo Carvalho (2024), essa escolha impacta diretamente a velocidade de convergência, a estabilidade do treinamento e a utilização de recursos computacionais. A Figura 4.9 representa como um *batch size* divide os dados de um *dataset*. Nesta figura, o *dataset* possui 10 “linhas”, isto é, 10 amostras. No caso 1, com *batch size*=1, o *dataset* foi agrupado em lotes de 1 amostra, totalizando 10 *batches*. No caso 2, com *batch size*=10, o *dataset* foi agrupado em lotes de 10 amostras, totalizando 1 *batch*. No caso 3, com *batch size*=2, o *dataset* foi agrupado em lotes de 2 amostras, totalizando 5 *batches*.

Figura 4.9: *Batch size* em um *dataset*.

Fonte: Elaboração dos autores (2025).

Em cada *batch*, é calculada a saída da rede, o erro médio da rede e são atualizados os pesos. Assim, dos casos representados na Figura 4.9, no Caso 1 os pesos são atualizados 10 vezes por época, no Caso 2 os pesos são atualizados 1 vez por época e no Caso 3 os pesos são atualizados 5 vezes por época.

Segundo Nelson (2023), existem três abordagens principais relacionadas à escolha do tamanho do lote:

- **Stochastic Gradient Descent (SGD):** corresponde ao caso em que o *batch size* é igual a 1, ou seja, os pesos são atualizados a cada amostra. Este método tende a ser mais rápido em iterações, mas introduz bastante ruído no processo de treinamento, podendo dificultar a convergência estável. Segundo Carvalho (2024) essa abordagem é vantajosa pois requer menos memória, tornando-a adequada, escalável e eficiente para conjuntos de dados grandes que não cabem na memória de uma vez. Em termos teóricos, quando o projetista não especifica o tamanho do lote, assume-se *batch size* = 1, caracterizando o SGD. Contudo, nos *frameworks* de uso prático, o valor padrão varia: no *Keras/TensorFlow* o *batch size* padrão é 32, enquanto no *PyTorch* é 1. Na Figura 4.9 esta abordagem é representada pelo Caso 1.
- **Batch Gradient Descent:** utiliza todo o conjunto de dados antes de realizar uma atualização dos pesos (*batch size* = número de amostras). Segundo Carvalho (2024),

esta abordagem é adequada para conjuntos de dados de menor escala, mas menos eficiente para dados com maior volume, visto que todo o conjunto de treinamento é processado de uma vez. Embora proporcione um gradiente estável, pode ser computacionalmente custoso e exigir grande quantidade de memória. Na Figura 4.9 esta abordagem é representada pelo Caso 2. É importante ressaltar que se o *batch size* for maior que o *dataset*, o treino se comporta como se o *batch size* fosse igual ao número total de amostras.

- **Mini-batch Gradient Descent:** utiliza lotes intermediários de dados (por exemplo, $batch\ size = 2, 16, 32, 64$ ou 128 amostras). Essa é a abordagem mais empregada na prática, pois equilibra a eficiência computacional e a estabilidade do processo de treinamento. Na Figura 4.9 esta abordagem é representada pelo Caso 3. Vale salientar que o número de *batches* não precisa ser divisor exato do *dataset*, e neste caso, o último *batch* será menor, mas a rede continua funcionando normalmente. Outro ponto importante é que, apesar de ser comum a escolha da extensão do *batch size* ser uma potência de base 2, não é um fato obrigatório. Entretanto, além desta escolha ser padrão de prática comum, Kandel e Castelli (2020) afirmam que esta prática se mostra eficaz para um treinamento mais rápido e eficiente, além de aproveitar ao máximo o processamento da Unidade de Processamento Gráfico (GPU).

Segundo Kandel e Castelli (2020), em geral, *batch sizes* pequenos podem convergir mais rápido, acelerando o treinamento por época, mas podem gerar oscilações no processo de minimização da função de custo. Por outro lado, *batch sizes* muito grandes fornecem um gradiente mais próximo do valor verdadeiro, atingindo mínimos ótimos que um lote pequeno não consegue atingir, porém, podem reduzir a capacidade de generalização da rede e demandar mais tempo de processamento e memória. Assim, a escolha do *batch size* adequado depende do problema em questão, do tamanho do conjunto de dados e da infraestrutura computacional disponível.

4.3.5 Otimizadores e Taxa de Aprendizagem

No treinamento de RNAs, o processo de ajuste dos pesos da rede ocorre por meio de algoritmos de otimização, também chamados de otimizadores. De acordo com Nelson (2023), a otimização é um belo e rico campo da matemática que fornece vários métodos

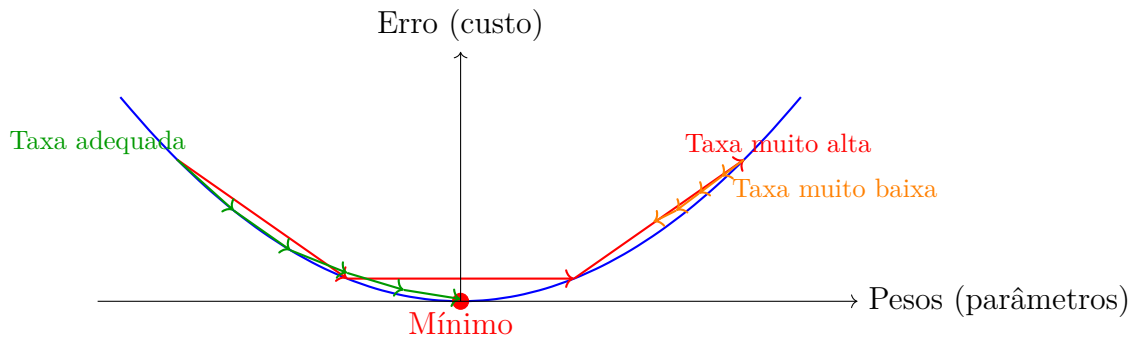
para localizar eficientemente a solução ótima, melhor, máxima, mínima ou extrema. Esses algoritmos utilizam os gradientes calculados na etapa de retropropagação para atualizar os parâmetros da rede, buscando minimizar a função de custo (NELSON, 2023). Entre os otimizadores mais utilizados, destacam-se o SGD, *RMSProp*, *Adagrad* e o *Adam*.

Um aspecto essencial no funcionamento de qualquer otimizador é a definição da *taxa de aprendizagem* (*learning rate*). Esse hiperparâmetro controla o tamanho do passo dado em cada atualização dos pesos. Segundo Furtado (2019), se a taxa for grande, pode divergir e se for pequena, pode tornar o processo muito lento, por isso, valores típicos da taxa de aprendizagem (η) situam-se entre $0,05 \leq \eta \leq 0,1$. Este autor menciona ainda a possibilidade de iniciar com um valor maior no começo do treinamento para acelerar a convergência e usar um valor menor no final para garantir um ajuste fino. Assim, a escolha adequada da taxa de aprendizagem é crucial para a eficiência e estabilidade do processo de otimização.

Além disso, é importante destacar que a escolha do *batch size* também está relacionada à taxa de aprendizagem, uma vez que ambos influenciam diretamente o processo de otimização. Segundo Kandel e Castelli (2020), o tamanho do lote e a taxa de aprendizado têm impacto um sobre o outro e podem ter um impacto enorme no desempenho da rede. Assim, a definição adequada desses hiperparâmetros é essencial para equilibrar velocidade de treinamento, estabilidade e capacidade de generalização do modelo.

Dentre os algoritmos mais empregados atualmente, destaca-se o *Adaptive Moment Estimation* (Adam). Esse otimizador utiliza estimativas dos momentos de primeira ordem (médias dos gradientes) e de segunda ordem (médias dos quadrados dos gradientes), o que o torna robusto e eficiente, principalmente em problemas que envolvem grandes volumes de dados e modelos com muitos parâmetros. Por essas características, o Adam tornou-se uma escolha padrão em diversas aplicações práticas de aprendizado profundo.

Figura 4.10: Influência da taxa de aprendizagem no processo de otimização.



Fonte: Elaboração dos autores (2025).

A Figura 4.10 ilustra, de forma esquemática, como diferentes valores de taxa de aprendizagem podem influenciar o processo de otimização. Nota-se que uma taxa muito alta pode fazer com que o modelo oscile e não convirja, enquanto uma taxa muito baixa pode tornar o processo lento. Já uma taxa adequada conduz o modelo de forma estável até o mínimo da função de custo.

4.3.6 *Momentum*

O aprendizado em uma MLP é geralmente realizado por meio do algoritmo de *backpropagation*, que ajusta os pesos da rede a cada iteração na direção que minimiza a função de custo. Para aprimorar a eficiência desse processo, é comum incluir um hiperparâmetro adicional denominado *momentum*, que pondera a influência das alterações dos pesos entre iterações sucessivas (SILVA, SPATTI E FLAUZINO, 2016). Considerando os neurônios da L -ésima camada, temos:

$$W_{ji}^{(L)}(t+1) = W_{ji}^{(L)}(t) + \underbrace{\alpha \cdot (W_{ji}^{(L)}(t) - W_{ji}^{(L)}(t-1))}_{\text{termo de momentum}} + \underbrace{\eta \cdot \delta_j^{(L)} \cdot Y_i^{(L-1)}}_{\text{termo de aprendizagem}}$$

onde α é definida como taxa de *momentum* e seu valor está compreendido entre zero e um. Furtado (2019) afirma que normalmente é recomendado, para treinamento de redes MLP, o *momentum* compreendido entre $0 \leq \alpha \leq 0.9$.

De acordo com Silva, Spatti e Flauzino (2016), a utilização do *momentum* torna o processo de convergência da rede mais eficiente pois se leva em consideração o critério de quão afastada está a solução atual da solução ótima. Por exemplo, um valor de

$\alpha = 0.9$ significa que 90% do gradiente anterior contribuirá para a atualização, enquanto o gradiente instantâneo da função de custo contribuirá com os 10% restantes. Carvalho (2024) afirma que este hiperparâmetro auxilia na aceleração do gradiente descendente em direções favoráveis, e suaviza a trajetória em direção ao mínimo ao integrar a direção dos gradientes anteriores ao cálculo atual, fazendo com que o algoritmo evite mínimos locais ou gradientes próximos de zero.

4.4 Aplicação de RNAs na resolução de problemas

Segundo Furtado (2019), são necessárias três fases para aplicar as RNAs à resolução de um problema. São elas: o treinamento, o teste e aplicação.

4.4.1 Treinamento

O treinamento é uma parte fundamental da RNA. É no treinamento que a rede ajusta os pesos e *bias*, resolvendo um conjunto de padrões de saída associados a padrões de entrada. Se a saída obtida no modelo não coincide com a saída desejada, os pesos e *bias* serão alterados proporcionalmente aos seus sinais de entrada (FURTADO, 2019). Este processo é repetido sequencialmente para todas as amostras de treinamento. Segundo Silva, Spatti e Flauzino (2016), o modelo só é considerado totalmente treinado quando o MSE entre duas épocas sucessivas for inferior à precisão, requerida ao problema a ser mapeado.

Além disso, vale salientar que segundo Furtado (2019),

“Para uma Rede Neural Artificial, o conceito de treinamento diferencia-se do conceito de aprendizado, pois este último está associado a uma tarefa que a rede está executando em função do treinamento, da sua arquitetura e da sua topologia. Já o treinamento é o processo de ensinar a RNA.”

Dessa forma, o treinamento é a etapa responsável por capacitar a rede para reconhecer padrões, preparando-a para ser avaliada posteriormente na fase de teste.

4.4.2 Teste

Na fase de teste, são apresentados padrões de entrada à rede, e as saídas obtidas são comparadas às saídas desejadas. Como menciona Furtado (2019), o teste funciona como uma avaliação para verificar se a rede está devidamente treinada, isto é, se reconhece adequadamente os padrões apresentados. Uma forma de fazer o teste é apresentar todo

o conjunto de dados e quando alguma entrada não for reconhecida, o teste pode ser interrompido, os pesos ajustados e o treinamento retomado. Outra proposta é realizar o teste por amostragem, onde na fase inicial, ao selecionar os dados, reserva-se uma parte dos dados (por exemplo 20%), conforme a preferência do projetista do modelo, para serem apenas dados de teste, sendo esta uma forma mais rápida e eficaz para a fase de teste, quando se tem um grande número de dados.

De acordo com Nelson (2023), medir o desempenho de modelos que preveem valores numéricos, como modelos de regressão, é mais simples do que modelos de classificação, porque há muitas maneiras de calcular distâncias entre números. Independentemente do tipo de problema, é importante, na fase de testes, decidir a(s) métrica(s) a ser(em) utilizada(s) para que o modelo tenha a precisão desejada.

Para problemas de classificação, é possível avaliar a saída do modelo em:

Verdadeiro Positivo (VP), **Verdadeiro Negativo (VN)**, **Falso Positivo (FP)** ou **Falso Negativo (FN)**.

Entre as métricas mais utilizadas em problemas de classificação, Nelson (2023) destaca:

- **Acurácia:** Porcentagem de vezes que o modelo de previsão acertou a classificação.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN}$$

- **Especificidade:** Avalia a proporção de instâncias negativas corretamente classificadas como negativas.

$$\text{Especificidade} = \frac{VN}{VN + FP}$$

- **Sensibilidade:** Avalia a proporção de instâncias positivas corretamente classificadas como positivas.

$$\text{Sensibilidade} = \frac{VP}{VP + FN}$$

Conforme afirmam Kandel e Castelli (2020), a escolha da métrica de avaliação deve estar alinhada ao objetivo do modelo, pois métricas como acurácia, sensibilidade e especificidade podem levar a interpretações distintas do desempenho da rede.

Além disso, talvez não seja o objetivo do projetista que a RNA reconheça 100% dos casos apresentados, mas sim um percentual satisfatório. Caso a métrica de desem-

penho utilizada não esteja num valor adequado ou esperado pelo projetista, retoma-se o treinamento (FURTADO, 2019). Mas, se a métrica utilizada já demonstra êxito, segue-se adiante para a fase final.

4.4.3 Aplicação

Com a rede treinada e já testada, temos a aplicação da RNA. Nessa fase não é feito nenhum tipo de ajuste nos parâmetros internos da rede. Nesta fase, o modelo pronto pelas fases anteriores é processado com o objetivo de gerar as saídas da rede para a resolução de problemas.

As RNAs são aplicadas nas mais diversas áreas do conhecimento. Silva, Spatti e Flauzino (2016) destacam três classes de problemas que concentram grande parte de suas aplicações: os problemas de classificação (ou reconhecimento) de padrões, os problemas de aproximação de funções e aqueles relacionados a sistemas dinâmicos (variantes no tempo). Manzan (2016) ressalta que essas aplicações se beneficiam da capacidade das redes neurais de identificar relações complexas entre variáveis de entrada e saída, o que as torna ferramentas versáteis e eficazes em diferentes domínios.

Diante do exposto, é importante salientar que atualmente, as RNAs desempenham um papel fundamental em diversas áreas e trazem avanços significativos. De acordo com Kandel e Castelli (2020), estudos recentes apontam que RNAs têm se mostrado eficazes em tarefas que vão desde classificação de padrões até previsão em sistemas dinâmicos, evidenciando sua versatilidade em diversos domínios. Além disso, de acordo com esses mesmos autores, a capacidade de generalização e a escolha adequada das métricas de avaliação são aspectos essenciais para garantir o desempenho confiável desses modelos. No campo da saúde, por exemplo, elas podem ser utilizadas para auxiliar no diagnóstico de doenças, identificar padrões em exames médicos e até prever surtos de epidemias (Rodrigues-Jr, Oliveira e Oliveira Jr., 2021). Com o avanço das tecnologias e o crescimento exponencial da quantidade de dados disponíveis, as RNAs se mostram cada vez mais eficientes. Esses aspectos reforçam nossa motivação para aplicar RNAs neste trabalho.

5 Redes Neurais Artificiais: aplicações básicas

Com a base teórica agora estabelecida, conforme destacado por Feitosa (2024), a prática emerge como o elemento crucial para consolidar o aprendizado e avançar no campo da tecnologia. A aplicação contínua e a exploração em projetos reais são, portanto, essenciais para que o conhecimento se transforme em habilidade. Mediante o exposto, após o estudo teórico sobre as RNAs, para compreender mais profundamente seu funcionamento, neste capítulo exploramos sua aplicação prática por meio de três experimentos com complexidades crescentes. O primeiro se trata da elaboração de uma RNA capaz de converter temperaturas de Celsius para Fahrenheit. Em seguida, foi desenvolvida uma segunda RNA para predição da nota de um aluno. A terceira e última aplicação básica, foi elaborada para identificar quais alunos fariam prova de Intervenção Pedagógica. Ambas as implementações foram realizadas no ambiente Google Colab, utilizando a linguagem de programação *Python*.

5.1 Modelo de RNA para conversão de temperaturas: Celsius para Fahrenheit

Duas escalas de temperatura muito utilizadas são Celsius e Fahrenheit. A escala Celsius, proposta por Anders Celsius (Suécia), define $0^{\circ}C$ para o ponto de congelamento da água e $100^{\circ}C$ para o ponto de ebulição. Já a escala Fahrenheit, proposta por Daniel Gabriel Fahrenheit (Polônia), define $32^{\circ}F$ como o ponto de congelamento e $212^{\circ}F$ como o de ebulição da água.

A conversão entre essas duas escalas é obtida pela função afim:

$$F = 1,8C + 32$$

onde F representa a temperatura em Fahrenheit e C a temperatura em Celsius.

Embora essa conversão seja resolvida facilmente por uma equação de 1º grau, optou-se por utilizá-la como exemplo inicial¹ para mostrar, de forma didática, o processo de construção, treinamento e avaliação de uma RNA. A seguir, descrevemos passo a passo sobre a implementação realizada. O código completo pode ser consultado no **Apêndice A.1**.

A seguir, detalharemos a elaboração do código, analisando cada célula individualmente.

Célula 01 – Importação das bibliotecas e definição dos dados de treino

Segundo Feitosa (2024), a linguagem *Python* possui um “ecossistema robusto” de bibliotecas que facilitam o desenvolvimento de soluções em IA. Nesse exemplo, foram utilizadas duas bibliotecas principais:

- *TensorFlow: framework* poderoso para o desenvolvimento de modelos de *deep learning*;
- *NumPy*: essencial para operações matemáticas e manipulação de *arrays* (vetores).

Para simplificar a escrita do código, convencionou-se o uso das abreviações *tf* para *TensorFlow* e *np* para *NumPy*.

Além disso, foram definidos os conjuntos de entrada (em Celsius) e de saída (em Fahrenheit), conforme a Tabela 5.1. Ao todo, foram utilizadas sete amostras para o treinamento do modelo.

Tabela 5.1: Temperaturas utilizadas para treinamento da RNA.

Celsius	-40	-10	0	8	15	22	38
Fahrenheit	-40	14	32	46	59	72	100

Fonte: Elaboração dos autores (2025).

Os valores foram armazenados em *arrays*, estruturas de dados próprias para manipulação numérica. Para garantir precisão, foi utilizado o tipo *float*, que suporta números

¹Essa primeira RNA foi inspirada no vídeo disponível em: https://youtu.be/iX_on3VxZzk?si=TCVDB6ADBZRMETI6. Acesso em: 08 ago. 2024

decimais. Também foram realizados testes com o tipo *int*, mas o resultado final da predição não sofreu alteração significativa.

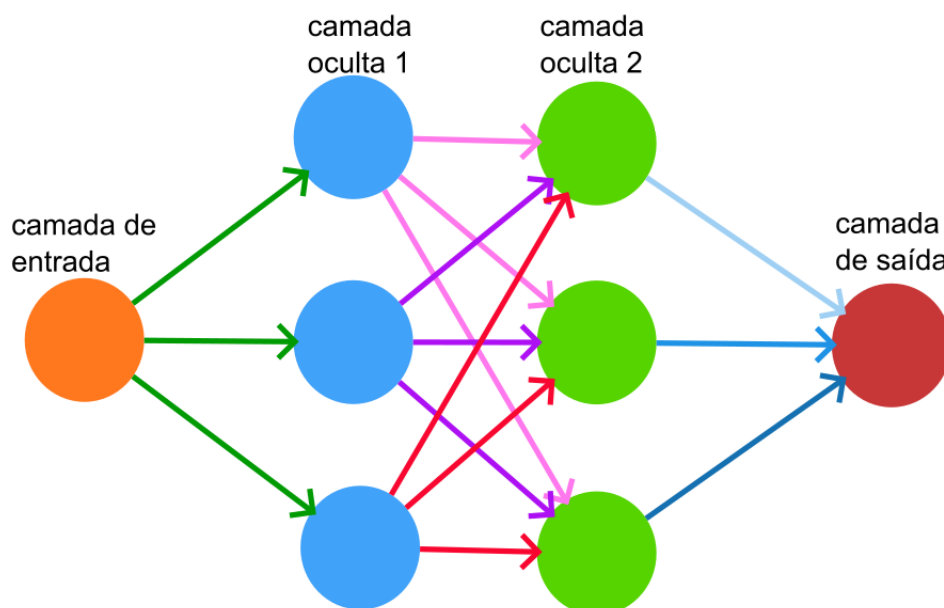
Célula 02 – Estrutura da RNA

Nesta etapa, foi definida a arquitetura da RNA. O modelo criado possui:

- Uma camada de entrada com 1 neurônio, isto é, um único atributo: temperatura em Celsius;
- Duas camadas ocultas, cada uma com 3 neurônios;
- Uma camada de saída com 1 neurônio (a temperatura em Fahrenheit).

A rede foi definida como densa pois todos os neurônios se conectam com os da camada anterior. Uma representação da estrutura dessa RNA é apresentada na Figura 5.1.

Figura 5.1: Representação esquemática da RNA para conversão de temperaturas



Fonte: Elaboração dos autores (2025).

Célula 03 – Compilação do modelo

Antes do treinamento, a rede neural precisa ser compilada. Para isso, foram definidos:

- **Otimizador:** Adam, escolhido por sua eficiência e rapidez, configurado com taxa de aprendizado de 0,1 (considerada relativamente alta);

- **Função de custo:** MSE, adequada para problemas de regressão.

Célula 04 – Treinamento do modelo

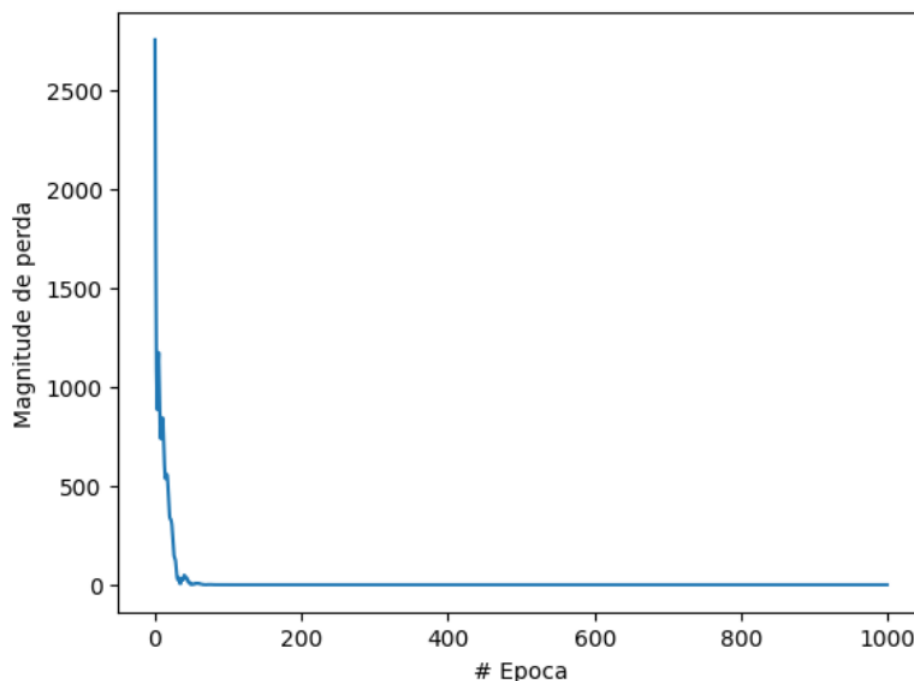
O treinamento foi realizado com 1000 épocas, ou seja, 1000 iterações sobre o conjunto de dados. Embora um número elevado de épocas possa melhorar o aprendizado, ele também torna o processo mais demorado. No caso, o treinamento levou cerca de 50 segundos.

Durante o processo, a rede buscou aprender a relação entre Celsius e Fahrenheit, ajustando pesos e (*bias*) por meio do algoritmo de retropropagação.

Célula 05 – Visualização do erro

Para acompanhar o desempenho do modelo, foi utilizado o pacote *matplotlib.pyplot* (abreviado *plt*). O gráfico gerado exibe no eixo *x* o número de épocas e, no eixo *y*, a magnitude do erro, como pode ser visto na Figura 5.2.

Figura 5.2: Variação do erro com até 1000 épocas.



Fonte: Elaboração dos autores (2025).

Os resultados mostraram que, após certo número de épocas, o erro se estabiliza, indicando que nem mesmo 1000 iterações seriam necessárias para alcançar alta precisão.

Célula 06 – Predição

Para validar o modelo, foi solicitado que a RNA previsse a conversão de $100^{\circ}C$. O resultado obtido foi $211,74744^{\circ}F$, valor muito próximo do real ($212^{\circ}F$), evidenciando a eficácia do treinamento.

Célula 07 – Pesos e *bias*

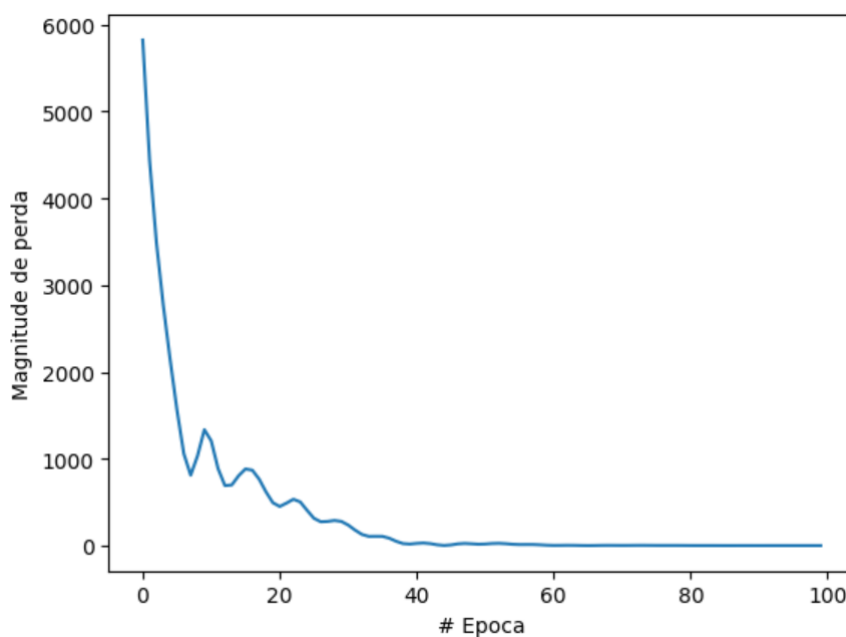
Por fim, foram analisados, apenas para efeito demonstrativo, os pesos e *bias* da rede. Os valores obtidos após a execução representam os melhores parâmetros encontrados durante o processo de treinamento e foram ajustados para compor o modelo final. Ressalta-se que o volume de dados é grande, pois, em redes densas, cada conexão possui um peso associado e cada neurônio possui um valor de *bias*.

Discussão sobre o número de épocas

Para efeito comparativo, também foi realizado o treinamento com apenas 100 épocas, o que reduziu o tempo de execução para cerca de 7 segundos. Nesse caso, a predição foi de aproximadamente $211,47624^{\circ}F$, ainda muito próxima do valor real. Isso demonstra que, para esse problema específico, não há necessidade de um número tão elevado de épocas.

A Figura 5.3 mostra o gráfico de erro considerando apenas 100 épocas.

Figura 5.3: Variação do erro com até 100 épocas.



Fonte: Elaboração dos autores (2025).

Conclui-se que, por se tratar de uma relação linear simples, o uso de funções de ativação não se fez necessário neste exemplo. Entretanto, em problemas mais complexos (que envolvem relações não lineares) essas funções tornam-se fundamentais para o desempenho do modelo.

5.2 Modelo de RNA para prever a nota de um aluno

No ano de 2025, o componente curricular de Matemática é ministrado para turmas do 8º ano do ensino fundamental. Uma dessas turmas é reduzida, composta por apenas 10 alunos, todos com laudos médicos, como Síndrome de Down, TDAH (Transtorno de Déficit de Atenção e Hiperatividade), transtorno de ansiedade, deficiência intelectual, deficiência intelectual com impacto na aprendizagem e dislexia. Em função dessas condições, os alunos apresentam dificuldades que os impedem de acompanhar uma turma regular.

Como menciona Silva (2023), não é possível igualar todos os estudantes utilizando os mesmos recursos e serviços, o mesmo currículo e os mesmos sistemas de avaliação. Por isso, para atender à heterogeneidade, são elaboradas quatro versões diferentes de cada atividade e avaliação. A turma conta também com o apoio de um professor de AEE, que auxilia três alunos durante as atividades, mas não durante as avaliações.

Assim, outra aplicação básica que elaboramos foi com dados reais de alunos dessa turma, sendo esses dados, referentes ao início do segundo trimestre até a primeira avaliação, aplicada em 10/07/2025. Os dados utilizados foram organizados em um *dataset* no *Excel*, apresentado na Tabela 5.2, cujas colunas são detalhadas na Tabela 5.3.

Tabela 5.2: *Dataset* utilizado para treinamento e teste da RNA

Revisão	NOTAS 1ºtrimestre	TAREFAS	FALTAS	NOTA PROVA
1	24.9	28	2	7.9
1	26.6	25	3	7.8
1	21.2	34	0	4.8
1	28.5	30	1	6.0
0	27.1	28	1	5.1
0	18.8	26	6	2.4
1	27.8	25	1	8.0
1	26.6	34	0	5.5
1	24.9	28	5	7.2
1	23.3	25	10	8.0

Fonte: Elaboração dos autores (2025).

Tabela 5.3: Descrição das colunas do *dataset*

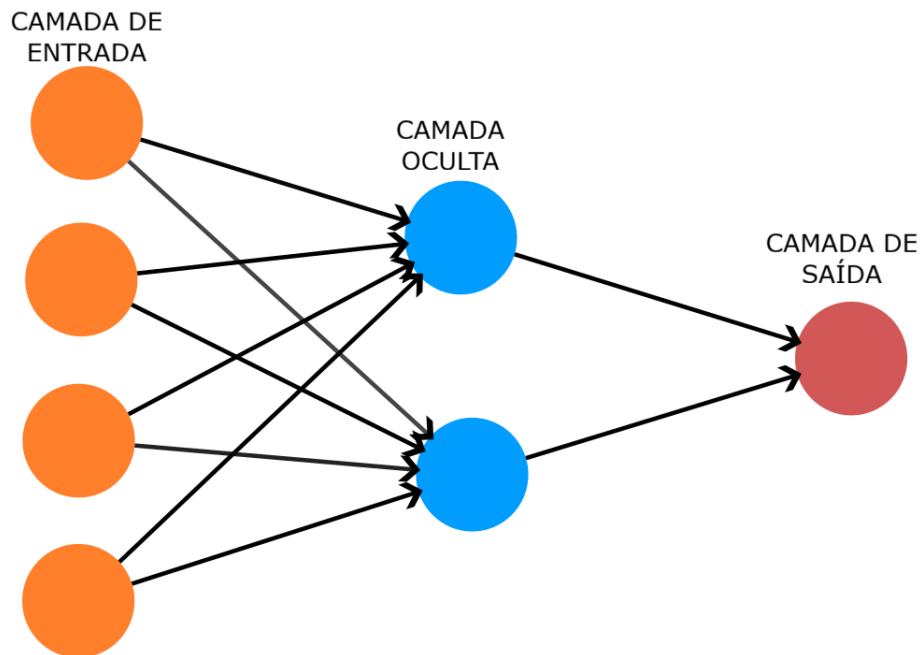
Coluna	Significado	Tipo de dado
Revisão	Fez a atividade de revisão semelhante à prova (0 = não, 1 = sim)	Binário
NOTAS 1ºtrimestre	Nota final do 1º trimestre de 2025	Numérico (0–30)
TAREFAS	Quantidade de tarefas entregues ao longo das aulas	Numérico (0–34)
FALTAS	Número de faltas no 2º trimestre até a data da avaliação	Numérico (0–35)
NOTA PROVA	Nota obtida na primeira avaliação do 2º trimestre	Regressão (0–8)

Fonte: Elaboração dos autores (2025).

A partir dos dados do *dataset*, elaboramos uma RNA para prever a nota dos alunos na avaliação. O *dataset* utilizado é reduzido e, portanto, os resultados obtidos têm caráter exploratório e ilustrativo. É importante destacar que o objetivo principal foi compreender o funcionamento da RNA e não necessariamente alcançar previsões altamente precisas.

Através do *dataset*, é notório que temos 4 atributos (sendo eles os dados referentes às quatro primeiras colunas) e assim, a RNA deve conter 4 neurônios na camada de entrada. Acrescentamos uma camada oculta com 2 neurônios e, por fim, uma camada de saída com 1 neurônio, resultando na arquitetura esquematizada na Figura 5.4.

Figura 5.4: Arquitetura da RNA para prever a nota de um aluno

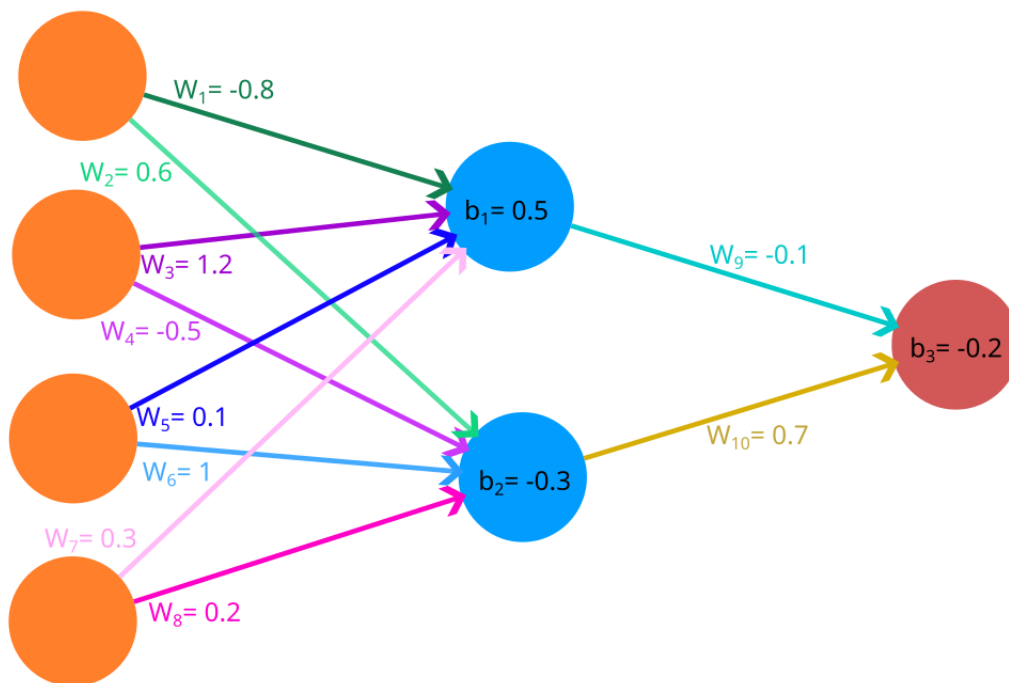


Fonte: Elaboração dos autores (2025).

Antes de programar o código no *Google Colab*, realizamos manualmente a simulação de uma época para compreender melhor o processo de cálculo em uma RNA. Na Figura 5.5 ilustramos os parâmetros iniciais (pesos e *bias*), escolhidos aleatoriamente.

Optou-se por realizar os cálculos de forma manual e escalar (isto é, operando termo a termo) em vez de matricialmente. Essa escolha foi feita com o objetivo de simplificar a compreensão do funcionamento interno da rede, permitindo visualizar passo a passo como cada peso e *bias* influencia o resultado final. Embora, na prática, as operações em RNAs sejam realizadas de forma matricial por questões de eficiência computacional, a abordagem manual facilita o entendimento conceitual do processo de propagação direta (*feedforward*).

Figura 5.5: Arquitetura da RNA para previsão da nota de um aluno, exibindo os pesos (W) e *bias* (b) atribuídos de forma aleatória antes do processo de treinamento



Fonte: Elaboração dos autores (2025).

Utilizando os valores da Figura 5.5 e o *dataset* da Tabela 5.2, calculamos as saídas da rede para os 8 primeiros alunos (80% da amostra destinada ao treino). Os resultados estão dispostos na Tabela 5.4.

Tabela 5.4: Cálculos manuais da primeira época da RNA.

Aluno	Camada Oculta	Saída
1	$\tilde{y}_1 = 1 \cdot -0.8 + 24.9 \cdot 1.2 + 28 \cdot 0.1 + 2 \cdot 0.3 + 0.5 = 32.98$ $\tilde{y}_2 = 1 \cdot 0.6 + 24.9 \cdot -0.5 + 28 \cdot 1 + 2 \cdot 0.2 - 0.3 = 16.25$	$\tilde{y} = \text{linear}(32.98 \cdot -0.1 + 16.25 \cdot 0.7 - 0.2) = 7.877$
2	$\tilde{y}_1 = 1 \cdot -0.8 + 26.6 \cdot 1.2 + 25 \cdot 0.1 + 3 \cdot 0.3 + 0.5 = 35.02$ $\tilde{y}_2 = 1 \cdot 0.6 + 26.6 \cdot -0.5 + 25 \cdot 1 + 3 \cdot 0.2 - 0.3 = 12.6$	$\tilde{y} = \text{linear}(35.02 \cdot -0.1 + 12.6 \cdot 0.7 - 0.2) = 5.118$
3	$\tilde{y}_1 = 1 \cdot -0.8 + 21.6 \cdot 1.2 + 34 \cdot 0.1 + 0 \cdot 0.3 + 0.5 = 29.02$ $\tilde{y}_2 = 1 \cdot 0.6 + 21.6 \cdot -0.5 + 34 \cdot 1 + 0 \cdot 0.2 - 0.3 = 23.5$	$\tilde{y} = \text{linear}(29.02 \cdot -0.1 + 23.5 \cdot 0.7 - 0.2) = 13.348$
4	$\tilde{y}_1 = 1 \cdot -0.8 + 28.5 \cdot 1.2 + 30 \cdot 0.1 + 1 \cdot 0.3 + 0.5 = 37.2$ $\tilde{y}_2 = 1 \cdot 0.6 + 28.5 \cdot -0.5 + 30 \cdot 1 + 1 \cdot 0.2 - 0.3 = 16.25$	$\tilde{y} = \text{linear}(37.2 \cdot -0.1 + 16.25 \cdot 0.7 - 0.2) = 7.455$
5	$\tilde{y}_1 = 0 \cdot -0.8 + 27.1 \cdot 1.2 + 28 \cdot 0.1 + 1 \cdot 0.3 + 0.5 = 36.12$ $\tilde{y}_2 = 0 \cdot 0.6 + 27.1 \cdot -0.5 + 28 \cdot 1 + 1 \cdot 0.2 - 0.3 = 14.95$	$\tilde{y} = \text{linear}(36.12 \cdot -0.1 + 14.95 \cdot 0.7 - 0.2) = 6.653$
6	$\tilde{y}_1 = 0 \cdot -0.8 + 18.8 \cdot 1.2 + 26 \cdot 0.1 + 6 \cdot 0.3 + 0.5 = 27.46$ $\tilde{y}_2 = 0 \cdot 0.6 + 18.8 \cdot -0.5 + 26 \cdot 1 + 6 \cdot 0.2 - 0.3 = 17.5$	$\tilde{y} = \text{linear}(27.46 \cdot -0.1 + 17.5 \cdot 0.7 - 0.2) = 9.304$
7	$\tilde{y}_1 = 1 \cdot -0.8 + 27.8 \cdot 1.2 + 25 \cdot 0.1 + 1 \cdot 0.3 + 0.5 = 35.86$ $\tilde{y}_2 = 1 \cdot 0.6 + 27.8 \cdot -0.5 + 25 \cdot 1 + 1 \cdot 0.2 - 0.3 = 11.6$	$\tilde{y} = \text{linear}(35.86 \cdot -0.1 + 11.6 \cdot 0.7 - 0.2) = 4.334$
8	$\tilde{y}_1 = 1 \cdot -0.8 + 26.6 \cdot 1.2 + 34 \cdot 0.1 + 0 \cdot 0.3 + 0.5 = 35.02$ $\tilde{y}_2 = 1 \cdot 0.6 + 26.6 \cdot -0.5 + 34 \cdot 1 + 0 \cdot 0.2 - 0.3 = 21$	$\tilde{y} = \text{linear}(35.02 \cdot -0.1 + 21 \cdot 0.7 - 0.2) = 10.998$

Fonte: Elaboração dos autores (2025).

Em seguida, comparamos a saída calculada \tilde{y} com a saída real y de cada aluno. Adotamos como função de custo o MSE. Os cálculos correspondentes estão organizados na Tabela 5.5.

Tabela 5.5: Cálculo manual do erro MSE na primeira época

Aluno	Saída real y	Saída calculada na primeira época \tilde{y}	Erro $y - \tilde{y}$	Erro quadrático $(y - \tilde{y})^2$
1	7.9	7.877	0.023	0.000529
2	7.8	5.118	2.682	7.193124
3	4.8	13.348	-8.548	73.068304
4	6.0	7.455	-1.455	2.117025
5	5.1	6.653	-1.553	2.411809
6	2.4	9.304	-6.904	47.665216
7	8.0	4.334	3.666	13.439556
8	5.5	10.998	-5.498	30.228004

Fonte: Elaboração dos autores (2025).

Substituindo os valores da Tabela 5.5 na fórmula do MSE, obtemos:

$MSE =$

$$\frac{1}{8}(0.000529+7.193124+73.068304+2.117025+2.411809+47.665216+13.439556+30.228004) =$$

$$= \frac{176.123567}{8} = 22.015445875$$

O resultado evidencia um erro elevado, o que era esperado, já que a rede ainda não havia sido ajustada. Ressalta-se que, embora o cálculo manual seja trabalhoso e demorado, computacionalmente o processo é simples e rápido.

Após compreender o funcionamento manual de uma época, passamos à implementação computacional da RNA no *Google Colab*, utilizando a linguagem de programação *Python* e a biblioteca *TensorFlow/Keras*. O *dataset* da Tabela 5.2 foi dividido em dois subconjuntos: 80% dos dados foram utilizados para treinamento e 20% para teste. A descrição do código completo elaborado no *Google Colab* pode ser visualizado no **Apêndice A.2**.

A arquitetura da rede foi definida conforme o esquema da Figura 5.4, composta por:

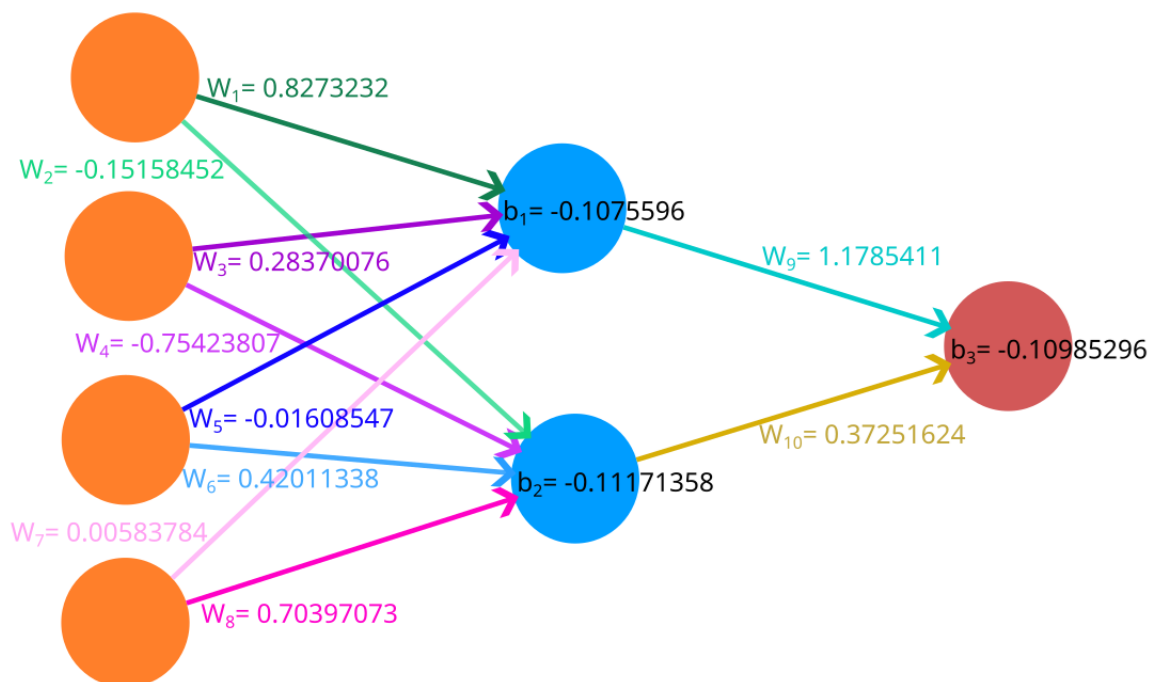
- **Camada de entrada:** 4 neurônios (um para cada atributo de entrada).
- **Camada oculta:** 2 neurônios totalmente conectados, com função de ativação ReLU.

- **Camada de saída:** 1 neurônio, com o uso da função de ativação linear, retornando a nota prevista. A função linear foi escolhida na saída porque o problema é de regressão, ou seja, a rede precisa retornar valores contínuos dentro da escala de notas.

O modelo foi compilado com o otimizador *Adam* (taxa de aprendizado de 0.001) e a função de perda MSE. Durante o treinamento, os pesos e *bias* foram ajustados pelo algoritmo de retropropagação (*backpropagation*), a fim de minimizar a diferença entre as saídas reais e as previstas pela rede.

Embora tenham sido definidas 5000 épocas, o treinamento foi interrompido após aproximadamente 350 épocas em função do critério de parada *early stopping*. Ao final, os valores ajustados de pesos e *bias* foram armazenados no modelo, conforme ilustrado na Figura 5.6.

Figura 5.6: Arquitetura da RNA com pesos e *bias* ajustados após o treinamento.



Fonte: Elaboração dos autores (2025).

Para avaliar o desempenho, foram utilizadas as amostras destinadas ao teste. O modelo obteve um MSE de aproximadamente 3.283, valor significativamente menor em comparação ao erro da primeira época calculada manualmente (22.015). Essa redução evidencia a capacidade de aprendizado da rede ao ajustar seus pesos e *bias*. No entanto, o erro final ainda pode ser considerado elevado em relação à escala das notas (0 a 8), o

que se deve principalmente ao número reduzido de amostras disponíveis no conjunto de dados, limitando a capacidade de generalização do modelo. Assim, os resultados devem ser interpretados como exemplos ilustrativos do funcionamento da RNA, mais do que como previsões confiáveis para situações reais.

Comparando as Figuras 5.5 e 5.6, observa-se a diferença entre os parâmetros aleatórios da primeira época e os valores ajustados após o treinamento (mais de 350 épocas). Essa evolução evidencia o processo de aprendizado da rede, isto é, a capacidade de modificar gradualmente pesos e *bias* para reduzir o erro e melhorar as previsões.

Em síntese, enquanto o cálculo manual permitiu compreender conceitualmente o funcionamento da rede e a influência de cada parâmetro nas saídas, a implementação computacional demonstrou a eficiência do treinamento automatizado, possibilitando que centenas de épocas fossem processadas em poucos segundos, algo inviável em cálculos manuais.

5.3 Modelo de RNA para identificar alunos que realizarão Intervenção Pedagógica

Após a apresentação de modelos de RNAs aplicados a problemas de regressão, esta seção tem como objetivo exemplificar o uso de uma RNA em um problema de classificação binária. Para isso, foi desenvolvida uma aplicação simples utilizando dados reais de uma turma regular do ano de 2025, composta por 20 alunos com diferentes níveis de desempenho escolar.

Na instituição analisada, a Intervenção Pedagógica consiste em uma estratégia adotada para apoiar alunos que não atingiram a média mínima nos dois primeiros trimestres, visando reduzir a necessidade de recuperação anual ao final do ano letivo. De acordo com os critérios estabelecidos pela escola, o aluno é encaminhado para a Intervenção Pedagógica caso obtenha nota igual ou inferior a 36 pontos, considerando o total de 60 pontos distribuídos nos dois primeiros trimestres.

Embora a identificação desses alunos possa ser realizada de forma direta por meio de planilhas eletrônicas ou sistemas acadêmicos, a proposta desta aplicação não é substituir essas ferramentas, mas demonstrar, de maneira didática, como uma RNA pode ser treinada para realizar decisões de classificação a partir de dados previamente rotulados.

O *dataset* utilizado foi construído com base nas notas do primeiro e do segundo tri-

mestre, associadas à informação sobre a realização ou não da Intervenção Pedagógica. Esse conjunto de dados, apresentado na Tabela 5.6, foi utilizado no treinamento supervisionado da RNA.

Tabela 5.6: *Dataset* para treinar a RNA sobre Intervenção Pedagógica.

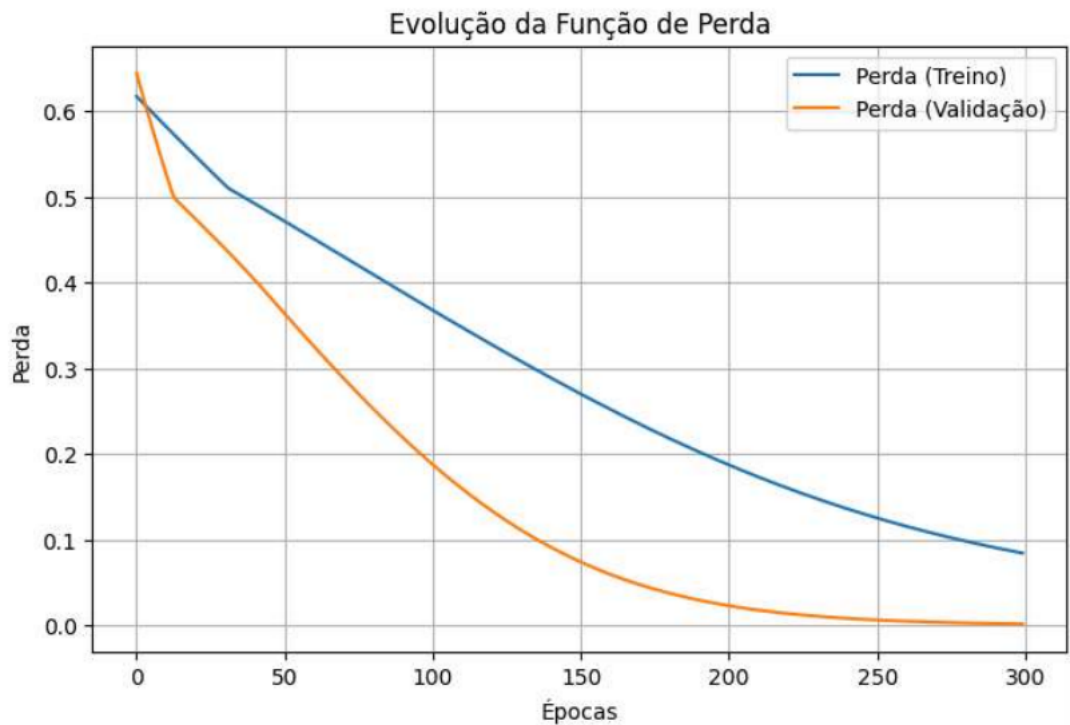
1ºTrimestre	2ºTrimestre	Ficou de intervenção?
30.0	30.0	NÃO
21.1	20.9	NÃO
20.4	18.5	NÃO
11.1	19.3	SIM
18.0	20.7	NÃO
30.0	30.0	NÃO
30.0	30.0	NÃO
18.3	18.0	NÃO
30.0	30.0	NÃO
22.5	24.1	NÃO
19.6	23.7	NÃO
16.6	18.4	SIM
30.0	30.0	NÃO
18.4	18.6	NÃO
24.7	25.4	NÃO
23.8	19.1	NÃO
22.1	27.3	NÃO
8.0	18.0	SIM
28.8	23.5	NÃO
20.9	26.0	NÃO

Fonte: Elaboração dos autores (2025).

A RNA foi configurada para receber como entradas as notas dos dois trimestres, resultando em dois neurônios na camada de entrada, e produzir uma saída binária indicando se o aluno realizaria ou não a Intervenção Pedagógica. Por se tratar de um problema de classificação, a saída do modelo assume valores discretos, representando as classes SIM ou NÃO.

A implementação completa do modelo, bem como os parâmetros utilizados no treinamento, encontra-se descrita no **Apêndice A.3**. Após o treinamento, o modelo apresentou acurácia de 100% e baixa perda, conforme ilustrado na Figura 5.7.

Figura 5.7: Evolução da Função Perda.



Fonte: Elaboração dos autores (2025).

Destaca-se que esse resultado está associado à simplicidade do problema e ao tamanho reduzido do *dataset*, motivo pelo qual esta aplicação possui caráter exclusivamente ilustrativo. Seu objetivo principal é evidenciar o funcionamento de uma RNA de classificação binária, complementando os exemplos de regressão apresentados nas seções anteriores.

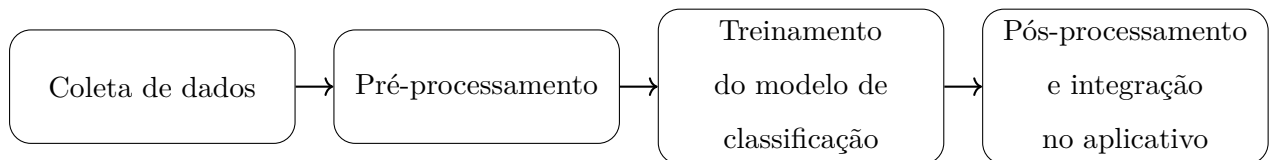
Por fim, foi incluída uma célula adicional que permite ao usuário inserir manualmente as notas de um aluno e verificar se, de acordo com o modelo treinado, ele realizaria ou não a Intervenção Pedagógica, reforçando o caráter demonstrativo da aplicação.

6 Modelo de RNA Proposto e Desenvolvimento do Aplicativo

Com o objetivo de criar um aplicativo como instrumento inicial para triagem de TEA, elaboramos um modelo de classificação binária baseado em uma RNA do tipo MLP, implementado por meio de técnicas de aprendizado profundo, para predição da probabilidade de TEA em adolescentes de 12 a 16 anos. O sistema foi projetado para ser acessível, eficiente e voltado ao público brasileiro.

A seguir, descrevemos as etapas de desenvolvimento do aplicativo.

Figura 6.1: Fluxograma das etapas para elaboração do aplicativo baseado em RNAs.



Fonte: Elaboração dos autores (2025).

6.1 Coleta de Dados

Para o treinamento do modelo proposto, utilizou-se um conjunto de dados sobre TEA em adolescentes, disponibilizado no repositório *UCI Machine Learning Repository* da Universidade da Califórnia em Irvine. O *dataset* foi originalmente apresentado por Tabtah (2017) e posteriormente atualizado por Artoni, Barbosa e Morandini (2022), contendo registros de indivíduos entre 12 e 16 anos avaliados por meio do questionário AQ-10¹.

O conjunto de dados é composto por 248 participantes, cada um descrito por 17

¹O AQ-10 (*Autism Spectrum Quotient – 10 items*) é uma versão reduzida do questionário AQ-50, desenvolvido por Allison *et al.* (2012), composta por dez itens com maior poder discriminativo para a identificação de traços do Transtorno do Espectro Autista. O instrumento é amplamente utilizado em contextos de triagem por apresentar alta sensibilidade e especificidade, além de maior facilidade de aplicação.

atributos. Desses, 10 correspondem a características comportamentais e sociais derivadas do questionário AQ-10, e 7 referem-se a informações pessoais, clínicas e operacionais.

A escolha do conjunto de 17 perguntas utilizadas neste trabalho fundamenta-se em critérios metodológicos, científicos e práticos. Inicialmente, as dez questões comportamentais adotadas correspondem ao questionário AQ-10 para adolescentes. Allison *et al.* (2012) afirmam que a seleção dos itens mais discriminativos permite manter elevados níveis de sensibilidade (0,93) e especificidade (0,95), mesmo com a redução significativa do número de perguntas, o que reforça a adequação do instrumento para fins de triagem inicial.

Além dessas dez questões, foram incorporadas sete características pessoais, clínicas e operacionais, amplamente utilizadas em bases de dados e aplicações computacionais de triagem de TEA, conforme proposto por Tabtah (2017) e Thabtah *et al.* (2018). Essas variáveis complementares contribuem para enriquecer o conjunto de entrada do modelo, aumentando sua capacidade preditiva sem comprometer a simplicidade do instrumento.

Ressalta-se que o objetivo do sistema desenvolvido é a triagem inicial de adolescentes com possíveis traços do Transtorno do Espectro Autista, e não o diagnóstico clínico. Nesse contexto, a utilização de um número reduzido de perguntas é desejável, pois favorece a usabilidade, a adesão dos usuários e a aplicação em ambientes digitais. Artoni, Barbosa e Morandini (2022) indicam, inclusive, que subconjuntos ainda menores de atributos podem alcançar elevados níveis de acurácia em modelos de aprendizado de máquina aplicados ao TEA, reforçando que a eficácia do modelo está mais relacionada à relevância das variáveis do que à sua quantidade.

As perguntas utilizadas no aplicativo TRIA-TEA Adolescente correspondem diretamente a esses 17 atributos, o que garante a compatibilidade entre o modelo treinado e o formulário apresentado aos usuários. Essas perguntas podem ser vistas na Tabela 6.1.

6.2 Pré-processamento

O pré-processamento teve como objetivo adequar os dados ao formato exigido pela RNA. As principais etapas consistiram em:

- tradução do conteúdo original para o idioma português;
- substituição dos valores binários 0 e 1 por “Não” e “Sim”, respectivamente;
- substituição das letras “m” e “f” por “Masculino” e “Feminino”;

Tabela 6.1: Perguntas correspondentes aos 17 atributos do conjunto de dados de triagem.

Perguntas	Possíveis respostas
1. Você está sempre percebendo padrões nas coisas?	Sim ou Não
2. Você, geralmente, se concentra mais na imagem geral do que nos pequenos detalhes?	Sim ou Não
3. Em um grupo social, você consegue acompanhar facilmente várias conversas diferentes ao mesmo tempo?	Sim ou Não
4. Se houver uma interrupção, você consegue voltar rapidamente ao que estava fazendo?	Sim ou Não
5. Frequentemente, você tem dificuldade em manter uma conversa em andamento?	Sim ou Não
6. Você se sai bem em pequenas conversas sociais?	Sim ou Não
7. Na infância, você costumava brincar de faz-de-conta com os amiguinhos?	Sim ou Não
8. Você apresenta dificuldades em se colocar no lugar de outra pessoa?	Sim ou Não
9. Você se sente à vontade em situações sociais?	Sim ou Não
10. Você tem dificuldade em fazer novos amigos?	Sim ou Não
11. Qual é sua idade?	12, 13, 14, 15 ou 16
12. Qual é o seu sexo?	Feminino ou Masculino
13. Qual é a sua etnia?	Asiático, Branco, Hispânico, Indígena, Latino, Negro, Oriente Médio, Sul Asiáticos ou Outros
14. Teve icterícia ao nascer?	Sim ou Não
15. País de residência?	Selecionar 1 dentre as 48 regiões disponíveis
16. Já usou algum aplicativo de triagem antes?	Sim ou Não
17. Quem está completando o teste?	Eu mesmo, Outros, Pai ou Mãe, Parente ou Profissional de saúde

Fonte: Elaboração dos autores (2025), com base em Tabtah (2017) e Artoni, Barbosa e Morandini (2022).

- normalização das variáveis numéricas no intervalo $[0,1]$.

O conjunto final de dados, com 248 instâncias e 17 atributos, foi dividido em 70% para o treinamento e 30% para validação. Esse processo garantiu a qualidade dos dados e permitiu avaliar o desempenho do modelo em diferentes subconjuntos.

6.3 Treinamento e Arquitetura do Modelo

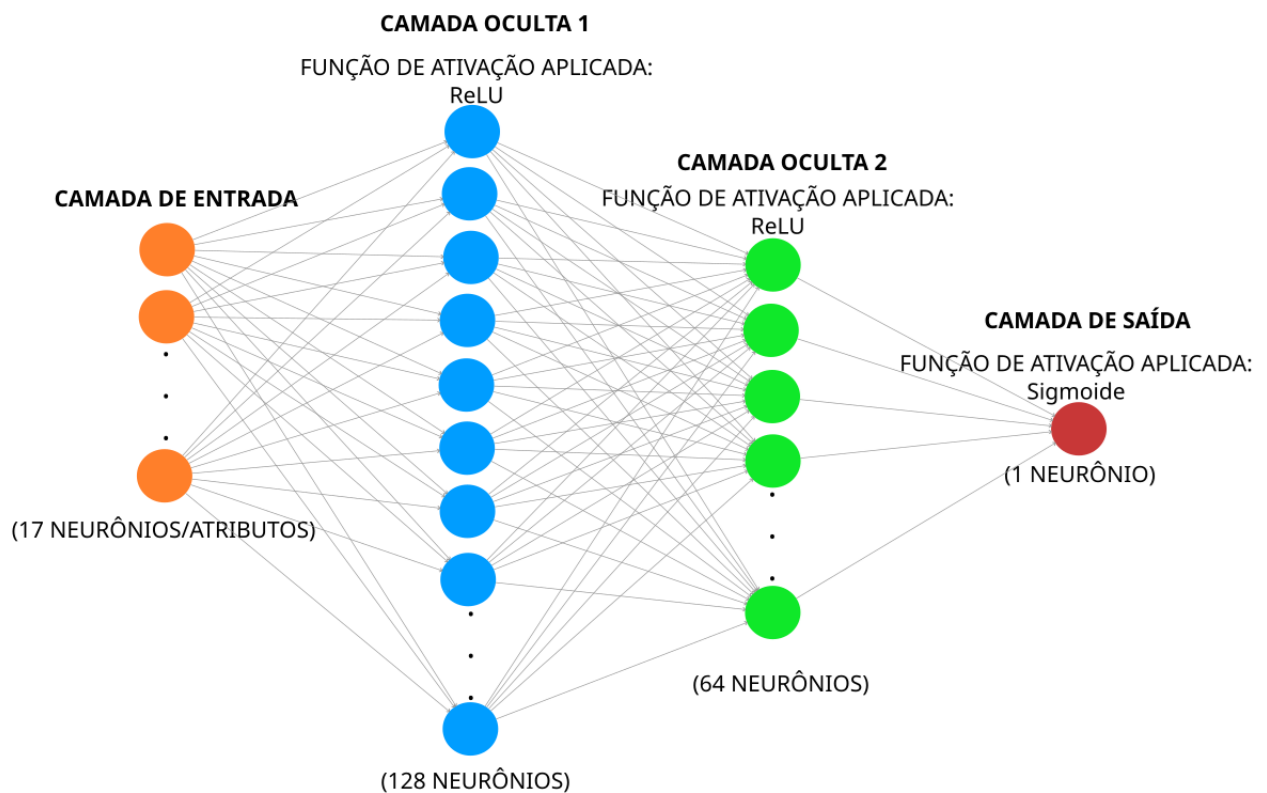
O modelo adotado é uma RNA do tipo MLP com arquitetura *feedforward*, composta por:

- **Camada de entrada:** 17 neurônios (atributos de entrada);
- **Duas camadas ocultas:** 128 e 64 neurônios, respectivamente, ambas com ativação ReLU e *dropout* de 30%²;
- **Camada de saída:** 1 neurônio com ativação sigmoide, responsável por produzir a probabilidade de TEA.

A Figura 6.2 ilustra a arquitetura dessa RNA. Embora as conexões entre os neurônios sejam totalmente densas, apenas uma parte delas foi representada para fins ilustrativos, visando facilitar a visualização da estrutura geral da rede.

²O *Dropout* é uma técnica de regularização que desativa aleatoriamente uma porcentagem dos neurônios durante o treinamento para prevenir o superajuste (“*overfitting*”). O número de parâmetros permanece constante.

Figura 6.2: Arquitetura da RNA usada para elaborar o aplicativo *web* Tria-TEA Adolescente.



Fonte: Elaboração dos autores (2025).

O modelo foi treinado com o otimizador *Adam* e função de custo MSE. Foram realizadas 40 épocas de treinamento, e o conjunto de validação foi usado para monitorar a perda e a acurácia a cada época.

Ao final do treinamento, o modelo apresentou desempenho satisfatório, com acurácia de 92% e $F1-Score$ ³ de 0,93, valores que demonstram a robustez da rede e a adequação do modelo à tarefa de triagem de TEA.

6.4 Pós-processamento e Integração

Após o treinamento, na camada de saída da RNA aplicamos a função de ativação Sigmoide, onde a saída \tilde{y} está no intervalo $[0,1]$, representando a probabilidade da saída. É fundamental que a saída do modelo seja clara para o usuário e, provavelmente, muitos usuários não entenderiam o resultado com probabilidade. Por isso, o resultado numérico

³Métrica harmônica que combina Precisão e Recall em um único valor, calculada pela média harmônica entre essas duas métricas ($F1 - Score = 2 \times (Precisão \times Sensibilidade) / (Precisão + Sensibilidade)$). Ideal para avaliar modelos em conjuntos de dados desbalanceados, pois equilibra a importância dos falsos positivos e falsos negativos.

foi convertido em quatro categorias interpretáveis pelo usuário:

- **Baixa probabilidade:** $0 \leq \tilde{y} < 0,25$
- **Probabilidade leve:** $0,25 \leq \tilde{y} < 0,5$
- **Probabilidade moderada:** $0,5 \leq \tilde{y} < 0,75$
- **Alta probabilidade:** $\tilde{y} \geq 0,75$

Essa conversão garante que a ferramenta não dê apenas um número, mas ofereça uma indicação clara da necessidade de acompanhamento. Além disso, permite uma interpretação intuitiva dos resultados, facilitando o uso por pais, educadores e profissionais de saúde. Cada resposta gera um código de identificação único, permitindo consultas posteriores com segurança e confidencialidade. A Figura 6.3 apresenta um exemplo de resultado gerado pelo aplicativo Tria-TEA Adolescente.

Figura 6.3: Exemplo de resultado gerado pelo Tria-TEA Adolescente.

Resultado da Avaliação

ID: 26/01/2026 09:21:21

Resultado: Probabilidade de TEA **alta**

Interpretação do resultado:

Com base nas informações fornecidas, a probabilidade de Transtorno do Espectro Autista aponta para uma ocorrência **muito provável**.

Faixas de classificação:

Baixa	< 0.25	Improvável
Leve	< 0.5	Possível
Moderada	< 0.75	Provável
Alta	≤ 1	Muito provável

O link a seguir é pessoal e intransferível. Guarde-o com segurança:

<https://apptea.onrender.com/predict?token=TEA12345&ID=26/01/2026 09:21:21>

Fonte: Elaboração dos autores (2025).

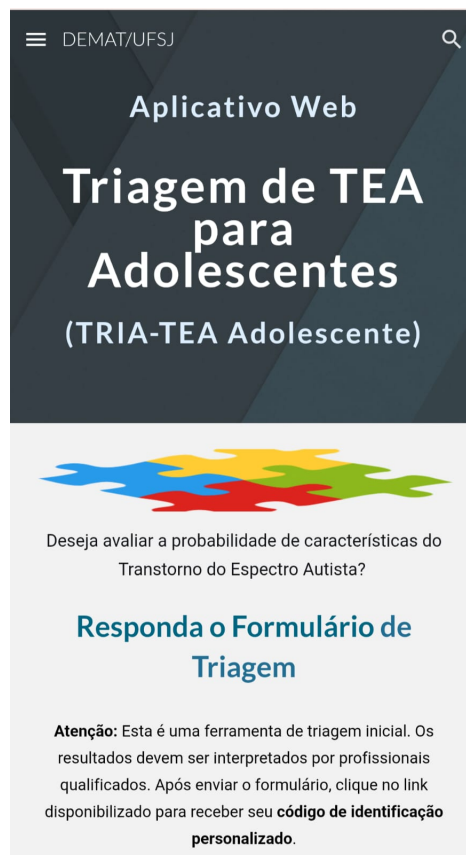
6.4.1 O Aplicativo Web TRIA-TEA Adolescente

O aplicativo *web* TRIA-TEA Adolescente foi desenvolvido em português e está disponível publicamente no endereço:

<https://sites.google.com/view/tea-ufsj>

A Figura 6.4 apresenta a página inicial do Aplicativo *web* Tria-TEA Adolescente.

Figura 6.4: Página inicial do aplicativo *web* Tria-TEA Adolescente.



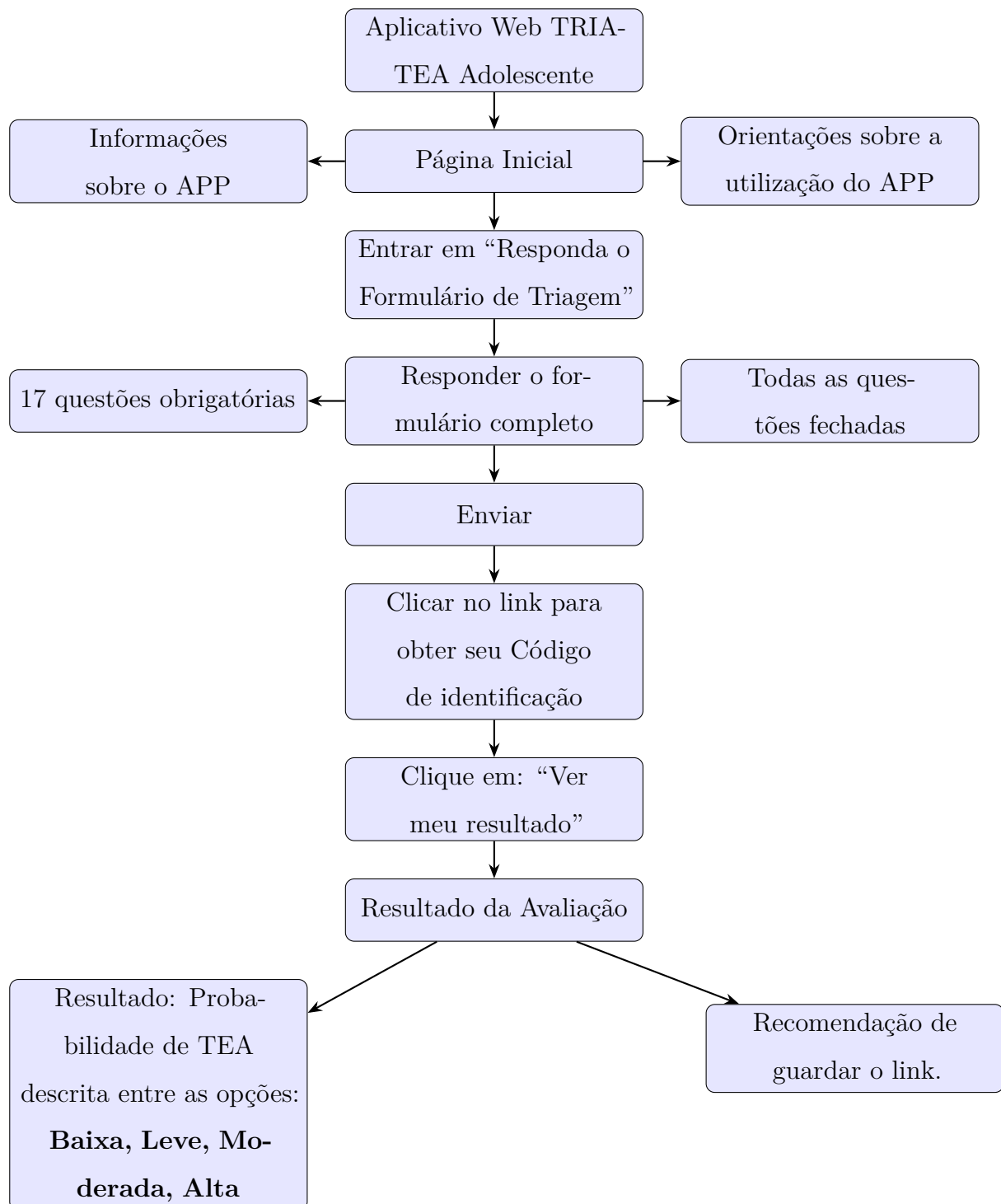
Fonte: Elaboração dos autores (2025).

O aplicativo foi desenvolvido em ambiente *web* utilizando e integrada ao Google Forms para coleta dos dados de triagem. Essa abordagem permitiu disponibilizar o sistema de forma gratuita e acessível a partir de qualquer dispositivo com acesso à internet, sem a necessidade de instalação.

O TRIA-TEA Adolescente apresenta uma interface simples e acessível, permitindo que o usuário responda às 17 questões de forma rápida. Após o envio, as respostas são processadas em tempo real pelo modelo de RNA, e o sistema exibe imediatamente a

estimativa de probabilidade de TEA. O TRIA-TEA Adolescente é de fácil compreensão, simples e rápido de preencher. A Figura 6.5 mostra o diagrama de navegação do aplicativo.

Figura 6.5: Diagrama de navegação do aplicativo proposto (TRIA-TEA Adolescente).



Fonte: Elaboração dos autores (2025).

6.5 Formulação Matemática do Modelo

Nesta seção, apresentamos a formulação matemática que descreve o funcionamento interno do modelo de RNA utilizado no aplicativo TRIA-TEA Adolescente. O objetivo é representar, de forma sucinta, o processo de transformação dos dados de entrada (as respostas do questionário) até a geração da saída (a probabilidade estimada de TEA).

Cada indivíduo é representado por um vetor de entrada \mathbf{x} , contendo os 17 atributos do questionário de triagem:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{17} \end{pmatrix}_{17 \times 1}$$

onde cada componente x_i representa uma resposta (ou característica) normalizada no intervalo $[0,1]$.

A RNA utilizada é do tipo MLP. A primeira camada oculta contém 128 neurônios, e cada conexão possui peso. Como a rede é densa, a matriz de peso da primeira camada oculta é do formato:

$$W^{(1)} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{117} \\ w_{21} & w_{22} & \cdots & w_{217} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1281} & w_{1282} & \cdots & w_{12817} \end{pmatrix}_{128 \times 17}$$

As operações realizadas em cada camada podem ser expressas matricialmente. Na primeira camada oculta, o vetor de entrada x é multiplicado pela matriz de pesos $\mathbf{W}^{(1)}$:

$$W^{(1)} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{117} \\ w_{21} & w_{22} & \cdots & w_{217} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1281} & w_{1282} & \cdots & w_{12817} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{17} \end{pmatrix} = \begin{pmatrix} w_{11} \cdot x_1 + w_{12} \cdot x_2 + \cdots + w_{217} \cdot x_{17} \\ w_{21} \cdot x_1 + w_{22} \cdot x_2 + \cdots + w_{217} \cdot x_{17} \\ \vdots \\ w_{1281} \cdot x_1 + w_{1282} \cdot x_2 + \cdots + w_{12817} \cdot x_{17} \end{pmatrix}$$

e, logo em seguida, somado ao vetor de *bias*. $\mathbf{b}_1 = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{17} \end{pmatrix}_{17 \times 1}$:

$$\mathbf{u}_1 = \mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}_1 = \begin{pmatrix} w_{11} \cdot x_1 + w_{12} \cdot x_2 + \cdots + w_{217} \cdot x_{17} + a_1 \\ w_{21} \cdot x_1 + w_{22} \cdot x_2 + \cdots + w_{217} \cdot x_{17} + a_2 \\ \vdots \\ w_{1281} \cdot x_1 + w_{1282} \cdot x_2 + \cdots + w_{12817} \cdot x_{17} + a_{17} \end{pmatrix}$$

Em seguida, o resultado é passado pela função de ativação ReLU (f_1), que mantém os valores positivos e zera os valores não positivos. O vetor resposta é a saída da primeira camada oculta (\tilde{y}_1)

$$\tilde{y}_1 = f_1(\mathbf{u}_1) = \max(0, \mathbf{u}_1)$$

Observação 7. O produto entre a matriz de peso e a matriz coluna (ou vetor coluna) é sempre possível nas RNAs densas pois o número de colunas da matriz de pesos de uma camada é igual ao número de saídas da camada anterior, isto é, da quantidade de linhas da matriz coluna que representa a saída da camada anterior.

Observação 8. O resultado do produto entre a matriz de peso e o vetor coluna é sempre um vetor coluna com quantidade de linhas definida pela quantidade de neurônios da camada em que está inserida. Como o vetor de *bias* também é um vetor coluna com a quantidade definida pela quantidade de neurônios dessa mesma camada, é possível fazer a soma entre o produto entre o vetor resultado do produto e o vetor *bias*.

O mesmo processo ocorre na segunda camada oculta, onde $\mathbf{W}^{(2)}$ representa a matriz de pesos que conecta as 128 saídas da primeira camada aos 64 neurônios da segunda camada.

$$W^{(2)} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1128} \\ w_{21} & w_{22} & \cdots & w_{2128} \\ \vdots & \vdots & \ddots & \vdots \\ w_{641} & w_{642} & \cdots & w_{64128} \end{pmatrix}_{64 \times 128}$$

Além disso, é somado o vetor coluna b_2 e é aplicada novamente a função de ativação

ReLU (f_2). Com \tilde{y}_2 representando a saída da segunda camada, temos:

$$\mathbf{u}_2 = \mathbf{W}^{(2)}\tilde{y}_1 + \mathbf{b}_2, \quad \tilde{y}_2 = f_2(\mathbf{u}_2) = \max(0, \mathbf{u}_2)$$

Por fim, a camada de saída realiza uma combinação linear dos 64 valores produzidos pela segunda camada, resultando em um único valor u_3 , que é transformado em uma probabilidade \tilde{y} por meio da função de ativação sigmoide (f):

$$u = \mathbf{W}^{(3)}\tilde{y}_2 + b, \quad \tilde{y} = f(u) = \frac{1}{1 + e^{-u}}$$

O valor \tilde{y} representa a probabilidade estimada de TEA. Valores próximos de 0 indicam baixa probabilidade, enquanto valores próximos de 1 indicam alta probabilidade de que o indivíduo apresente traços do espectro.

Em termos gerais, o conjunto de todos os parâmetros treináveis da rede (pesos e *bias*) pode ser agrupado em um vetor:

$$\boldsymbol{\alpha} = (\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}, \mathbf{b}_1, \mathbf{b}_2, b)^t \in \mathbb{R}^s$$

onde

$$s = \underbrace{(17 \times 128) + (128 \times 64) + (64 \times 1)}_{\text{Pesos}} + \underbrace{128 \times 1 + (64 \times 1) + 1}_{\text{bias}} = 10.625$$

é o número total de parâmetros do modelo. Além disso, cada componente de $\boldsymbol{\alpha}$ representa uma entrada líquida de um neurônio da rede, ou seja, o valor que cada neurônio recebe antes de passar pela função de ativação.

Para introduzir a não linearidade característica das redes neurais, aplicamos uma função de ativação genérica ξ sobre essas entradas líquidas, obtendo a saída de cada neurônio da camada:

$$u = \xi(\boldsymbol{\alpha}), \tag{6.1}$$

em que ξ representa qualquer função de ativação (como sigmoide, ReLU ou tangente hiperbólica), sem se restringir a uma específica.

Dessa forma, a saída final da rede pode ser escrita de maneira compacta como

$$\tilde{y} = f(\xi(\boldsymbol{\alpha})), \quad (6.2)$$

assim, temos que

$$\tilde{y} = f \circ \xi \quad (6.3)$$

ou, de forma ainda mais simplificada, de (6.2) e (6.3), temos:

$$\tilde{y} = \tilde{y}(\boldsymbol{\alpha}), \quad (6.4)$$

representando a função de predição aprendida pela rede durante o treinamento.

Em síntese, o modelo matemático apresentado expressa a transformação funcional realizada pela RNA: os dados de triagem passam por múltiplas camadas de combinações lineares e funções de ativação, até que o sistema produza, na saída, uma estimativa probabilística de TEA. Esse mapeamento é o núcleo do aplicativo TRIA-TEA Adolescente, pois traduz o aprendizado obtido a partir do conjunto de dados em uma ferramenta prática para triagem.

7 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo desenvolver um modelo computacional baseado em RNAs para a triagem inicial do TEA em adolescentes de 12 a 16 anos, resultando na criação do aplicativo *web* TRIA-TEA Adolescente. A partir do estudo teórico sobre o TEA, suas manifestações no contexto escolar e a importância do diagnóstico precoce, aliado à fundamentação matemática e computacional das RNAs, foi possível projetar e implementar um modelo funcional, com resultados satisfatórios em termos de acurácia, precisão e sensibilidade, além de reforçar a importância da Matemática Aplicada.

Os resultados obtidos demonstram que o modelo desenvolvido é capaz de identificar padrões comportamentais e cognitivos relacionados ao TEA de forma confiável, evidenciando o potencial das RNAs como ferramenta de apoio à triagem precoce. Além disso, a integração do modelo em um aplicativo *web* acessível e intuitivo reforça sua aplicabilidade prática, permitindo que pais, educadores e profissionais de saúde obtenham informações úteis de maneira rápida e segura.

A elaboração deste trabalho representou uma trajetória de intenso aprendizado e reflexão. Unir os campos da Matemática, da Computação e da Educação em torno de uma temática tão sensível e necessária como o TEA, possibilitou compreender a força das conexões entre teoria e prática.

Ao longo do processo, foi necessário revisitar fundamentos matemáticos, compreender o funcionamento das RNAs e, ao mesmo tempo, refletir sobre as práticas pedagógicas e as necessidades do público-alvo. Cada etapa desse percurso reforçou a importância da curiosidade, da pesquisa e da formação contínua.

Concluir este trabalho foi uma experiência profundamente significativa. Ele reflete o olhar de uma educadora que ama a Matemática e suas aplicações, que vê a tecnologia como aliada da inclusão e acredita no poder transformador do conhecimento. Este estudo também reafirma que, quando unimos rigor científico e propósito humano, a Matemática deixa de ser apenas uma ciência exata e se torna uma ponte entre o pensar e o cuidar.

Como trabalho futuro, pretende-se ampliar o TRIA-TEA e desenvolver modelos semelhantes para a triagem do TEA em crianças e adultos, considerando que já existem bases de dados disponíveis para essas faixas etárias, como as propostas por Tabtah (2017). Além disso, seria interessante, para um trabalho futuro, usar uma base de dados com amostras brasileiras e validar o desempenho do modelo em diferentes faixas etárias, de modo a aprimorar a precisão e a aplicabilidade do TRIA-TEA Adolescente.

Outra perspectiva relevante é integrar o uso de RNAs em aulas de Matemática (por exemplo, na conversão de temperaturas em conteúdos de função afim) promovendo uma abordagem interdisciplinar entre Ciência da Computação e Matemática.

Conclui-se, portanto, que este trabalho não apenas apresentou um modelo computacional eficiente para a triagem de TEA, mas também reafirmou a convicção de que a educação, quando aliada à ciência e à tecnologia, é capaz de transformar realidades e abrir caminhos para uma sociedade mais inclusiva, humana e consciente.

REFERÊNCIAS

- 1 SILVA, A. L. C.; FERNANDES, M. H. S. Transtorno do Espectro do Autismo (TEA): breve história para uma longa discussão. **Revista Master - Ensino, Pesquisa e Extensão**, v. 8, n. 15, 2023. Disponível em: <https://doi.org/10.47224/revistamaster.v8i15.252>. Acesso em: 15 out. 2024.
- 2 EVÊNCIO, K. M. M.; FERNANDES, G. P. História do Autismo: Compreensões Iniciais. **Id on Line Revista de Psicologia**, v. 13, n. 47, p. 133-138, 2019. Disponível em: <https://doi.org/10.14295/idonline.v13i47.1968>. Acesso em: 16 out. 2024.
- 3 MAIA, F. A.; ALMEIDA, M. T. C.; ALVES, M. R.; BANDEIRA, L. V. S. B.; SILVA, V. B.; NUNES, N. F.; CARDOSO, L. C. G.; SILVEIRA, M. F. Transtorno do espectro do autismo e idade dos genitores: estudo de caso-controle no Brasil, 2018. Disponível em: <https://doi.org/10.1590/0102-311X00109917>. Acesso em: 17 out. 2024.
- 4 LAVOR, M. D. L. S. S.; LOPES, C. N.; DAMACENO, M. M. D. P.; DA SILVA, L. A.; ALVES, C. G. C.; FILHO, F. C.; MENINO, M. E. G.; GUEDES, T. A. L. O autismo: aspectos genéticos e seus biomarcadores: uma revisão integrativa. **Brazilian Journal of Health Review**. v. 4, n. 1, p. 3274–3289, 2021. Disponível em: <https://doi.org/10.34119/bjhrv4n1-258>. Acesso em: 17 nov. 2024.
- 5 TABTAH, F. Autistic spectrum disorder screening data for adolescent [dataset]. UCI Machine Learning Repository, 2017. Disponível em: <https://archive.ics.uci.edu/dataset/420/autistic+spectrum+disorder+screening+data+for+adolescent>. Acesso em: 25 nov. 2024.
- 6 Thabtah, F. Kamalov, and K. Rajab. A new computational intelligence approach to detect autistic features for autism screening. *International Journal of Medical Informatics*, 117:112–124, 2018. Disponível em: [10.1016/j.ijmedinf.2018.06.009](https://doi.org/10.1016/j.ijmedinf.2018.06.009). Acesso em: 26 nov. 2024.
- 7 NICOLETTI, M. A.; HONDA, F. R. Transtorno do Espectro Autista: uma abordagem sobre as políticas públicas e o acesso à sociedade. **Infarma**.v. 33, n. 2, p. 117-130, 2021. Disponível em: <https://bdta.abcd.usp.br/directbitstream/b84b89b8-671a-4b01-99ff-0bb2a5d71fb9/>

3061585.pdf.. Acesso em: 15 jan. 2025.

8 JÚNIOR, F. P. Fita símbolo da conscientização a respeito do autismo. 2010. 988×1600 pixels. Disponível em:

https://pt.wikipedia.org/wiki/Fita_quebra-cabe%C3%A7as#/media/Ficheiro:Autismo-fita3.jpg. Acesso em: 04 abr. 2025.

9 MORAIS, I. R. V. de; SOARES, P. H. L.; KRELA, G. L. S.; SOUSA, E. G.; MIRANDA, J. A. G.; SERAFIM, A. C. P.; HOLNIK, M. F.; QUINTIERO, C. H. H.; BRITO, D. I. L. de; PINHEIRO, A. C. C.; OLIVEIRA, V. E. T. de; NONATO, J. W. B.; MARTINS, J. M.; CASTELHANO, H. R. de J. A.; TAVARES, D. O impacto do excesso de tempo de tela na primeira infância e sua relação com a predisposição ao autismo.

Brazilian Journal of Health Review, v. 9, n.2, p. 01-18, 2025. Disponível em:

<https://doi.org/10.34119/bjhrv8n2-023>. Acesso em 08 abr. 2025.

10 BARONI, I.; PADILHA, E. J. A Matemática por Trás das Redes Neurais. **Centro Universitário Internacional, UNINTER**, 2021. Disponível em:

<https://repositorio.uninter.com/bitstream/handle/1/1023/ITANIM%c3%81%20BARONI%20-%20RU%202607819.pdf?sequence=1&isAllowed=y>. Acesso em: 04 maio 2025.

11 COZMAN, F. G.; NERI, H. O que, afinal, é Inteligência Artificial? **Inteligência Artificial: Avanços e Tendências**. In: COZMAN, F. G.; PLONSKI, G. A.; NERI, H., São Paulo: IEA e C4AI-USP, p. 21-29, 2021. Disponível em:

<https://www.livrosabertos.abcd.usp.br/portaldelivrosUSP/catalog/download/650/579/2181?inline=1>. Acesso em: 04 maio 2025.

12 FINGER, M. Quando se compra inteligência artificial, o que de fato se leva para casa? Além do “oba-oba” **Inteligência Artificial: Avanços e Tendências**. In: COZMAN, F. G.; PLONSKI, G. A.; NERI, H., São Paulo: IEA e C4AI-USP, p. 97-111, 2021.

Disponível em:

<https://www.livrosabertos.abcd.usp.br/portaldelivrosUSP/catalog/download/650/579/2181?inline=1>. Acesso em: 04 maio 2025.

13 RODRIGUES-Jr., J. F.; OLIVEIRA, M. C. F.; OLIVEIRA Jr., O. N. O futuro da ciência e tecnologia com as máquinas inteligentes **Inteligência Artificial: Avanços e Tendências**. In: COZMAN, F. G.; PLONSKI, G. A.; NERI, H., São Paulo: IEA e C4AI-USP, p. 151-180, 2021. Disponível em:

<https://www.livrosabertos.abcd.usp.br/portaldelivrosUSP/catalog/download/650/579/2181?inline=1>. Acesso em: 04 maio 2025.

14 MAZZON, J. A.; BARROS, F. M. O. Aplicações de técnicas de Análise de Dados e Inteligência Artificial em Finanças e Marketing **Inteligência Artificial: Avanços e Tendências**. In: COZMAN, F. G.; PLONSKI, G. A.; NERI, H., São Paulo: IEA e

C4AI-USP, p. 373-403, 2021. Disponível em:

<https://www.livrosabertos.abcd.usp.br/portaldelivrosUSP/catalog/download/650/579/2181?inline=1>. Acesso em: 04 maio 2025.

15 FURTADO, M. I. V. Redes Neurais Artificiais: Uma Abordagem Para Sala de Aula. **Atena Editora**. v. 19, 2019. DOI: www.doi.org/10.22533/at.ed.262191504. Disponível em:

<https://educapes.capes.gov.br/bitstream/capes/432794/1/e-book-Redes-Neurais-Artificiais-uma-Abordagem-para-Sala-de-Aula.pdf> Acesso em: 07 maio 2025.

16 SEDESE-MG. 800x282 pixels. Disponível em:

<https://images.app.goo.gl/ABnU4>. Acesso em: 07 jun. 2025.

17 MANGABEIRA, G. L. Diagnóstico do Transtorno do Espectro Autista. **Transtorno do Espectro do Autismo: Um estudo epidemiológico no Norte de Minas Gerais..** In: MAIA, F. A.; SILVEIRA, M. F.; ARAÚJO, S. L. N. O., Scrito Sensu Editora, p. 16-18, 2022. Disponível em:

<https://doi.org/10.35170/ss.ed.9786586283709.02>. Acesso em: 08 jun. 2025.

18 MAIA, F. A. Transtorno do Espectro do Autismo: Uma revisão sobre etiologia, epigenética e mutação de novo. **Transtorno do Espectro do Autismo: Um estudo epidemiológico no Norte de Minas Gerais..** In: MAIA, F. A.; SILVEIRA, M. F.; ARAÚJO, S. L. N. O., Scrito Sensu Editora, p. 19-23, 2022. Disponível em:

<https://doi.org/10.35170/ss.ed.9786586283709.03>. Acesso em: 09 jun. 2025.

19 BANDEIRA, L. V. S. Aprendizagem e inclusão no Transtorno do Espectro do Autismo. **Transtorno do Espectro do Autismo: Um estudo epidemiológico no Norte de Minas Gerais..** In: MAIA, F. A.; SILVEIRA, M. F.; ARAÚJO, S. L. N. O., Scrito Sensu Editora, p. 46-49, 2022. Disponível em:

<https://doi.org/10.35170/ss.ed.9786586283709.09>. Acesso em: 10 jun. 2025.

20 COSTA, A. A.; ALMEIDA, M. M. T.; MAIA, F. A.; SILVEIRA, M. F. TEA e o uso de medicamentos, tabaco, álcool e drogas ilícitas durante a gestação. **Transtorno do Espectro do Autismo: Um estudo epidemiológico no Norte de Minas Gerais..** In: MAIA, F. A.; SILVEIRA, M. F.; ARAÚJO, S. L. N. O., Scrito Sensu Editora, p. 46-49, 2022. Disponível em:

<https://doi.org/10.35170/ss.ed.9786586283709.13>. Acesso em: 10 jun. 2025.

21 OLIVEIRA, L. M. M.; SILVEIRA, M. F.; MAIA, F. A. Transtorno do Espectro do Autismo e fatores pós-natais. **Transtorno do Espectro do Autismo: Um estudo epidemiológico no Norte de Minas Gerais..** In: MAIA, F. A.; SILVEIRA, M. F.; ARAÚJO, S. L. N. O., Scrito Sensu Editora, p. 76-80, 2022. Disponível em:

<https://doi.org/10.35170/ss.ed.9786586283709.16>. Acesso em: 11 jun. 2025.

- 22 Censo Demográfico 2022 : pessoas com deficiência e pessoas diagnosticadas com transtorno do espectro autista : resultados preliminares da amostra. **IBGE**, Rio de Janeiro, 2025. Disponível em:
<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/43464-censo-2022-identifica-2-4-milhoes-de-pessoas-diagnosticadas-com-autismo-no-brasil>. Acesso em: 12 jun. 2025.
- 23 SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. Redes Neurais Artificiais para engenharia e ciências aplicadas. **Artliber Editora**, e. 2, 2016. Disponível em:
https://artliber.com.br/amostra/redes_neurais.pdf. Acesso em: 21 jul. 2025.
- 24 FEITOSA, R. D. F. Introdução ao Python [livro eletrônico] **Cegraf UFG**, e. 1, 2024. Disponível em:
<https://doi.org/10.5216/FEL.int.ebook.978-85-495-0977-2/2024>. Acesso em: 28 jun. 2025.
- 25 MANZAN, J. R. G. Análise de desempenho de redes neurais artificiais do tipo multilayer perceptron por meio do distanciamento dos pontos do espaço de saída. 2016. 129 f. Tese (Doutorado em Ciências) - Universidade Federal de Uberlândia, Uberlândia, 2016. Disponível em:
<https://doi.org/10.14393/ufu.te.2016.133>. Acesso em: 01 set. 2025.
- 26 RIZZO, I. V.; CANATO, R. L. C. Inteligência Artificial: Funções de ativação. **Revista Prospectus**, v. 2, n. 2, p. 51-65, 2020. Disponível em:
<https://share.google/waAEwL8CfUHTJkBA>. Acesso em: 04 set. 2025.
- 27 ARTONI, A. A.; BARBOSA, C. R. S. C.; MORANDINI, M. Autism Spectrum Disorder Diagnosis Assistance using Machine Learning. **Revista de Informática Teórica e Aplicada**, v. 29 n. 03 p. 36–53, 2022. Disponível em:
<https://seer.ufrgs.br/index.php/rita/article/view/126309/87499>. Acesso em: 05 set. 2025.
- 28 ALLISON, C.; AUYEUNG, B.; BARON-COHEN, S. Toward Brief “Red Flags” for Autism Screening: The Short Autism Spectrum Quotient and the Short Quantitative Checklist in 1,000 Cases and 3,000 Controls. *J Am Acad Child Adolesc Psychiatry*, 51(2):202–212. e. 7, 2012. Disponível em:
[10.1016/j.jaac.2011.11.003](https://doi.org/10.1016/j.jaac.2011.11.003). Acesso em: 05 set. 2025.
- 29 KANDEL, I.; CASTELLI, M. The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. **ICT Express**, v.6, n. 4, p. 312-315, 2020. Disponível em:
<https://www.sciencedirect.com/science/article/pii/S2405959519303455>. Acesso em: 19 set. 2025.

- 30 CUGLER, E.; GARCIA, A. A. F. Afirmação de Trump de que Tylenol causa autismo é nova mentira estimulada por Robert F. Kennedy Jr. G1, 24 set. 2025 . <https://g1.globo.com/saude/noticia/2025/09/24/afirmacao-de-trump-de-que-tylenol-causa-autismo-e-nova-mentira-estimulada-por-robert-f-kennedy-jr.ghtml>. Acesso em: 26 set. 2025.
- 31 SILVA, A. N.. Prática pedagógica: desafios de transformar a teoria na práxis inclusiva. **Humanidades e Tecnologia (FINOM)**, v. 40, n. 1, p. 398-410, 2023. Disponível em: <https://core.ac.uk/download/pdf/578691608.pdf>. Acesso em: 02 out. 2025.
- 32 CARVALHO, R. S. Análise do desempenho de redes neurais artificiais em problemas de classificação multiclasse. Dissertação (Mestrado em Matemática) — Universidade de Brasília, Brasília, 2024. Disponível em: https://repositorio.unb.br/bitstream/10482/51941/1/RaquelSouzaCarvalho_DISSERT.pdf. Acesso em: 13 out. 2025.
- 33 BRASIL, R. M.; BRITO, D. F.; BARRETO, A. C. L. G.; JUNIOR, H. S. N. Fundamentos e aplicação de Redes Neurais Artificiais à saúde e humanidades: tutorial—parte 1. **Revista Presença**, v. 11, n. 26, p. 405-419, 2025. Disponível em: <https://ojs.celsotech.io/index.php/revista/article/view/51>. Acesso em: 14 out. 2025.
- 34 QUIRINO, G. S.; CASTRO, J. P.; SILVA, A. M. T. C.; SOBRINHO, H. M. da R. (2025). Atualizações sobre triagem, diagnóstico e manejo do transtorno do espectro autista. **Revista Eletrônica Acervo Saúde**, v. 25, e. 19469. Disponível em: <https://acervomais.com.br/index.php/saude/article/view/19469>. Acesso em: 23 out. 2025.
- 35 DA SILVA, A. A. S., DE SOUZA, C. R. T. (2023). CRIANÇAS DESATENTAS E HIPERATIVAS: A IMPORTÂNCIA DA COMUNICAÇÃO ENTRE PROFESSORES E PAIS. **Fundação Universidade Federal de Mato Grosso do Sul**. Disponível em: <https://repositorio.ufms.br/jspui/retrieve/1dbd9dd3-6cd5-4212-a65e-79e7a8201ff9/8643.pdf>. Acesso em: 05 nov. 2025.
- 36 MUTSCHLER, J. F. C.; SILVA, L. de M. N. G. ADOLESCÊNCIA: DESAFIOS, OBSTÁCULOS E POTENCIALIDADES NA CONSTRUÇÃO DE SI E DO MUNDO. **ARACÊ**, [S. l.], v. 7, n. 9, p. e8274, 2025. Disponível em: <https://periodicos.newsciencepubl.com/arace/article/view/8274>. Acesso em: 21 jan. 2026.
- 37 Nelson, H. Essential Math for AI **O’Reilly Media**, e.1, 2023.
- 38 STEWART, J. Cálculo, v.1, e.7. americana. São Paulo, Cengage Learning, 2013.
- 39 STEWART, J. Cálculo, v.2, e.7. americana. São Paulo, Cengage Learning, 2013.

- 40 LIMA, E. L. Análise real volume 1. Funções de uma variável. Rio de Janeiro: IMPA, v.2, e.8, 2006.
- 41 LIMA, E. L. Análise real volume 2. Funções de uma variável. Rio de Janeiro: IMPA, v.2, e.1, 2004.
- 42 LIMA, E. L. Análise real, v.3 : Análise vetorial . Funções de uma variável. Rio de Janeiro: IMPA, e.4, 2014.

APÊNDICE A

Códigos dos Modelos de RNA

A.1 Modelo de RNA para conversão de temperaturas

29/10/25, 15:33

Celsius para Fahrenheit - Colab

✓ Célula 01

```
#@title Célula 01

import tensorflow as tf
import numpy as np

celsius=np.array([-40, -10, 0, 8, 15, 22, 38], dtype=float)
fahrenheit=np.array([-40, 14, 32, 46, 59, 72, 100], dtype=float)
```

✓ Célula 02

```
#@title Célula 02

entrada=tf.keras.Input(shape=[1])
oculta1=tf.keras.layers.Dense(units=3)
oculta2=tf.keras.layers.Dense(units=3)
saida=tf.keras.layers.Dense(units=1)
modelo= tf.keras.Sequential([entrada, oculta1,oculta2,saida])
```

✓ Célula 03

```
#@title Célula 03

modelo.compile(
    optimizer=tf.keras.optimizers.Adam(0.1),
    loss='mean_squared_error')
```

v Célula 04

```
##@title Célula 04

print("Começando treinamento...")
registro= modelo.fit(celsius,fahrenheit,epochs=1000,verbose=False)
print("Modelo treinado")
```

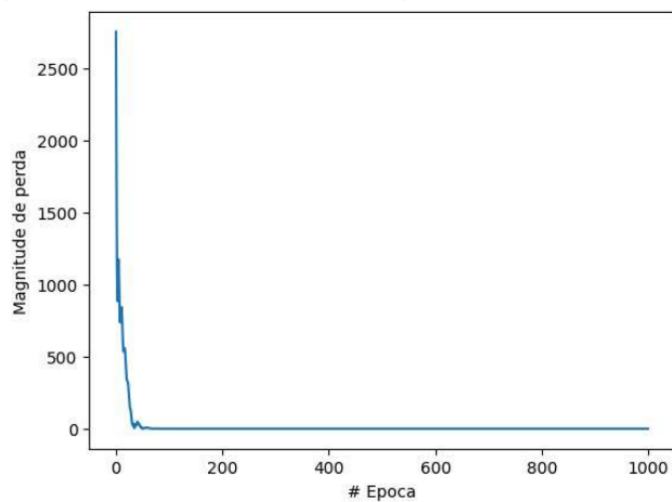
```
Começando treinamento...
Modelo treinado
```

v Célula 05

```
##@title Célula 05

import matplotlib.pyplot as plt
plt.xlabel("# Epoca")
plt.ylabel("Magnitude de perda")
plt.plot(registro.history["loss"])
```

[<matplotlib.lines.Line2D at 0x7f540d52de50>]




29/10/25, 15:33

Celsius para Fahrenheit - Colab

v Célula 06

```
#@title Célula 06

print("Façamos uma predição!")
resultado= modelo.predict(np.array([100.0]))
print("O resultado é " + str(resultado) + " fahrenheit!")
```

```
Façamos uma predição!
1/1  0s 54ms/step
O resultado é [[211.74744]] fahrenheit!
```

v Célula 07

```
#@title Célula 07

print("Variáveis internas do modelo")
print(oculta1.get_weights())
print(oculta2.get_weights())
print(saida.get_weights())
```

```
Variáveis internas do modelo
[array([[ -0.43139505,  0.87294525, -0.21368356]], dtype=float32), array([ 2.7379072,  3.4611228, -1.6540093], dtype=float32)
[array([[ -0.8624468 , -0.46142578,  1.0454037 ],
        [-1.0587033 , -0.52253544,  1.3199036 ]], dtype=float32), array([[ -3.4436307, -2.6681128,  3.4205825], dtype=float32)]
[array([[ -1.0470773 ],
        [-0.69768167],
        [ 1.2357863 ]], dtype=float32), array([3.277071], dtype=float32)]
```

A.2 Modelo de RNA para prever notas de alunos

30/09/25, 18:55

prever nota de um aluno - Colab

▼ Célula 01

```

#@title Célula 01

import tensorflow as tf
import pandas as pd
import numpy as np

```

▼ Célula 02

```

#@title Célula 02

from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True)

```

▼ Célula 03

```

#@title Célula 03

df=pd.read_excel('/content/drive/MyDrive/Dados_ALUNOS.xlsx')

```

▼ Célula 04

```

#@title Célula 04
df

```

	Revisão	NOTAS 1ºtrimestre	TAREFAS	FALTAS	NOTA PROVA
0	1	24.9	28	2	7.9
1	1	26.6	25	3	7.8
2	1	21.2	34	0	4.8
3	1	28.5	30	1	6.0
4	0	27.1	28	1	5.1
5	0	18.8	26	6	2.4
6	1	27.8	25	1	8.0
7	1	26.6	34	0	5.5
8	1	24.9	28	5	7.2
9	1	23.3	25	10	8.0

Próximas etapas: [Gerar código com df](#) [New interactive sheet](#)

▼ Célula 05

```

#@title Célula 05

y=df['NOTA PROVA']
x=df.drop('NOTA PROVA', axis=1)

```

▼ Célula 06

```

#@title Célula 06

x_treino, x_teste=x[0:8], x[8:]
y_treino, y_teste=y[0:8], y[8:]

```

▼ Célula 07

```

#@title Célula 07

x_treino.shape

(8, 4)

```

✓ Célula 08

```
##@title Célula 08

entrada=tf.keras.layers.Input(shape=[4])
oculta=tf.keras.layers.Dense(units=2)
saida=tf.keras.layers.Dense(units=1,activation='linear')
modelo= tf.keras.Sequential([entrada, oculta, saida])
```

✓ Célula 09

```
##@title Célula 09

modelo.compile(
    optimizer=tf.keras.optimizers.Adam(0.001),
    loss='mean_squared_error')
```

✓ Célula 10

```
##@title Célula 10

print("Começando treinamento...")

from tensorflow.keras.callbacks import EarlyStopping

early = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

resultado= modelo.fit(x_treino, y_treino,
    epochs=5000,
    batch_size=32,
    validation_data=(x_teste, y_teste),
    callbacks=[early],
    verbose=False)

print("Modelo treinado")
```

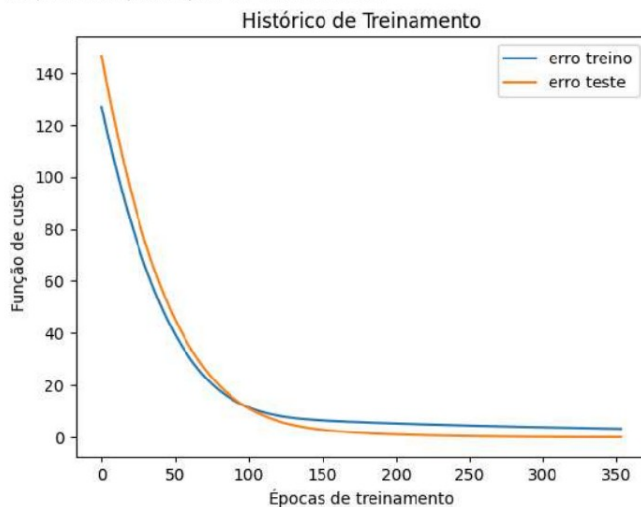
```
Começando treinamento...
Modelo treinado
```

✓ Célula 11

```
##@title Célula 11

import matplotlib.pyplot as plt
plt.plot(resultado.history["loss"])
plt.plot(resultado.history["val_loss"])
plt.title('Histórico de Treinamento')
plt.ylabel('Função de custo')
plt.xlabel('Épocas de treinamento')
plt.legend(['erro treino', 'erro teste'])
```

<matplotlib.legend.Legend at 0x7ff3b8b22ff0>



v Célula 12

#@title Célula 12

```
print("Façamos uma validação")
entrada_manual = np.array([[1, 23.3, 25, 10]])
```

```
validação = modelo.predict(entrada_manual)
print("O resultado é " + str(validação))
```

```
WARNING:tensorflow:6 out of the last 8 calls to <function TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_dis
Façamos uma validação
1/1 ----- 0s 47ms/step
O resultado é [[8.014002]]
```

v Célula 13

#@title Célula 13

```
for camada in modelo.layers:
    pesos, bias = camada.get_weights()
    print(f'Camada: {camada}')
    print('Pesos:\n', pesos)
    print('Bias:\n', bias)
    print('-----')
```

Camada: <Dense name=dense_10, built=True>

Pesos:

```
[[ 0.8273232 -0.15158452]
 [ 0.28370076 -0.75423807]
 [-0.01608547  0.42011338]
 [ 0.00583784  0.70397073]]
```

Bias:

```
[-0.1075596 -0.11171358]
-----
```

Camada: <Dense name=dense_11, built=True>

Pesos:

```
[[1.1785411 ]
 [0.37251624]]
```

Bias:

```
[-0.10985296]
-----
```

v Célula 14

#@title Célula 14

```
erro_mse = modelo.evaluate(x_treino, y_treino, verbose=0)
print(f'Erro final (MSE): {erro_mse}')
```

Erro final (MSE): 3.2828431129455566

A.3 Modelo de RNA para identificar alunos que realizarão Intervenção Pedagógica

26/01/26, 10:57

intervenção - Colab

▼ Célula 01

```
#@title Célula 01

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, confusion_matrix
import tensorflow as tf
from tensorflow.keras.callbacks import EarlyStopping
```

▼ Célula 02

```
#@title Célula 02

dataset = pd.read_excel('/content/drive/MyDrive/Dados-intervencao.xlsx')
dataset.head()
```

	1ºTrimestre	2ºTrimestre	Ficou de intervenção?
0	30.0	30.0	NÃO
1	21.1	20.9	NÃO
2	20.4	18.5	NÃO
3	11.1	19.3	SIM
4	18.0	20.7	NÃO

Próximas etapas: [Gerar código com dataset](#) [New interactive sheet](#)

▼ Célula 03

```
#@title Célula 03

X = dataset[['1ºTrimestre', '2ºTrimestre']]
y = dataset['Ficou de intervenção?'].map({'SIM': 1, 'NAO': 0})

# Remove rows with NaN values from X and y
nan_mask = y.isna()
X = X[~nan_mask]
y = y[~nan_mask]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

▼ Célula 04

```
#@title Célula 04

entrada = tf.keras.layers.Input(shape=[2]) # Change shape to 2 to match the number of features
oculta1 = tf.keras.layers.Dense(units=8, activation='relu')
oculta2 = tf.keras.layers.Dense(units=4, activation='relu')
saida = tf.keras.layers.Dense(units=1, activation='sigmoid')

modelo = tf.keras.Sequential([entrada, oculta1, oculta2, saida])
modelo.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Célula 05

```
#@title Célula 05

print("Iniciando treinamento...")

resultado = modelo.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs=300,
    batch_size=8,
    verbose=False
)

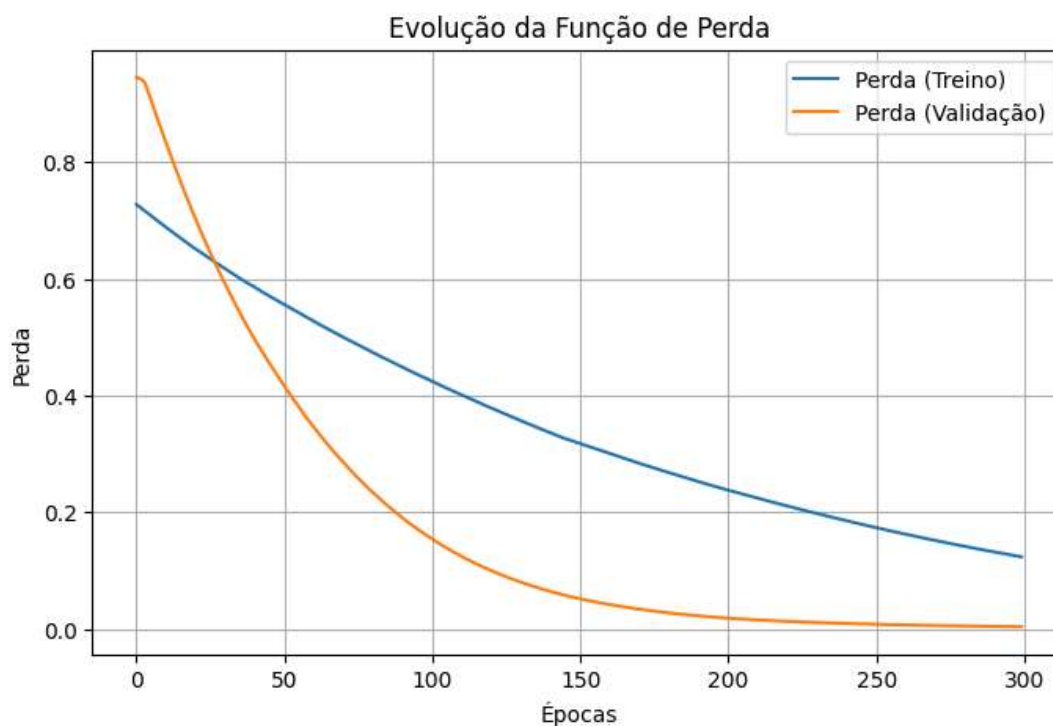
print("Treinamento concluído.")
```

```
Iniciando treinamento...
Treinamento concluído.
```

Célula 06

```
#@title Célula 06

plt.figure(figsize=(8, 5))
plt.plot(resultado.history["loss"], label='Perda (Treino)')
plt.plot(resultado.history["val_loss"], label='Perda (Validação)')
plt.title('Evolução da Função de Perda')
plt.xlabel('Épocas')
plt.ylabel('Perda')
plt.legend()
plt.grid(True)
plt.show()
```



Célula 07

```
#@title Célula 07

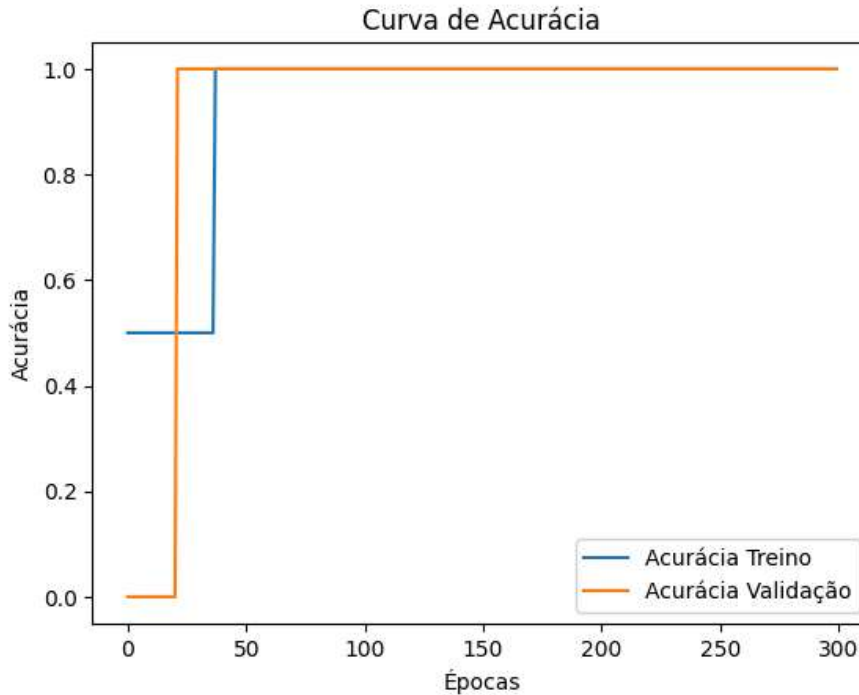
loss, acc = modelo.evaluate(X_test, y_test, verbose=0)
print(f"Acurácia no teste: {acc*100:.2f}%")
```

```
Acurácia no teste: 100.00%
```

✓ Célula 08

```
#@title Célula 08
```

```
plt.plot(resultado.history['accuracy'], label='Acurácia Treino')
plt.plot(resultado.history['val_accuracy'], label='Acurácia Validação')
plt.title('Curva de Acurácia')
plt.xlabel('Épocas')
plt.ylabel('Acurácia')
plt.legend()
plt.show()
```



✓ Célula 09

```
#@title Célula 09
```

```
# Célula 09 - Predição para novos dados
print("Exemplo de predição:")
```

```
nota1_pred = 10.0
nota2_pred = 22.0
notas_para_predicao = np.array([[nota1_pred, nota2_pred]])
```

```
notas_escaladas = scaler.transform(notas_para_predicao)
probabilidade_intervencao = modelo.predict(notas_escaladas)
classe_predita = (probabilidade_intervencao > 0.5).astype("int32")[0][0]
```

```
mapeamento_inverso = {0: 'NAO', 1: 'SIM'}
print(f"Para as notas {nota1_pred} e {nota2_pred}, a previsão é: {mapeamento_inverso[classe_predita]}")
```

Exemplo de predição:

1/1 ————— 0s 75ms/step

Para as notas 10.0 e 22.0, a previsão é: SIM

/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have warnings.warn(