

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS E TECNOLOGIA PARA SUSTENTABILIDADE
DEPARTAMENTO DE FÍSICA, QUÍMICA E MATEMÁTICA

Sara Fernanda de Jesus Mion

**A utilização de Redes Neurais Artificiais no estudo
da Constante de Acoplamento da
Cromodinâmica Quântica**

Sorocaba

2024

Sara Fernanda de Jesus Mion

**A utilização de Redes Neurais Artificiais no estudo
da Constante de Acoplamento da
Cromodinâmica Quântica**

Trabalho de Conclusão de Curso apresentado ao curso de Licenciatura Plena em Física para obtenção do título de Licenciado em Física.

Orientação: Prof. Dr. Renato Fernandes Cantão

Sorocaba

2024

Mion, Sara Fernanda de Jesus

A utilização de Redes Neurais Artificiais no estudo da
Constante de Acoplamento da Cromodinâmica Quântica /
Sara Fernanda de Jesus Mion -- 2024.
75f.

TCC (Graduação) - Universidade Federal de São Carlos,
campus Sorocaba, Sorocaba

Orientador (a): Renato Fernandes Cantão

Banca Examinadora: Sérgio Dias Campos, Johnny

Vilcarromero López

Bibliografia

1. Cromodinâmica Quântica. 2. Redes Neurais Artificiais.
3. Aprendizado de Máquina. I. Mion, Sara Fernanda de Jesus. II. Título.

Ficha catalográfica desenvolvida pela Secretaria Geral de Informática
(SIn)

DADOS FORNECIDOS PELO AUTOR

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



FUNDAÇÃO UNIVERSIDADE FEDERAL DE SÃO CARLOS

COORDENAÇÃO DO CURSO DE LICENCIATURA EM FÍSICA - SOROCABA - CCFL-So/CCTS
Rod. João Leme dos Santos km 110 - SP-264, s/n - Bairro Itinga, Sorocaba/SP, CEP 18052-780
Telefone: (15) 32298859 - <http://www.ufscar.br>

DP-TCC-FA nº 3/2024/CCFL-So/CCTS

Graduação: Defesa Pública de Trabalho de Conclusão de Curso

Folha Aprovação (GDP-TCC-FA)

FOLHA DE APROVAÇÃO

SARA FERNANDA DE JESUS MION

A UTILIZAÇÃO DE REDES NEURAS ARTIFICIAIS NO ESTUDO DA CONSTANTE DE ACOPLAMENTO DA CROMODINÂMICA QUÂNTICA

Trabalho de Conclusão de Curso

Universidade Federal de São Carlos - Campus Sorocaba

Sorocaba, 15 de fevereiro de 2024

ASSINATURAS E CIÊNCIAS

Cargo/Função	Nome Completo
Orientador	Prof. Dr. Renato Fernandes Cantão
Membro da Banca 1	Prof. Dr. Sérgio Dias Campos
Membro da Banca 2	Prof. Dr. Johnny Vilcarromero López



Documento assinado eletronicamente por **Renato Fernandes Cantao, Docente**, em 15/02/2024, às 15:48, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Sergio Dias Campos, Docente**, em 16/02/2024, às 07:44, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Johnny Vilcarromero Lopez, Chefe de Departamento**, em 16/02/2024, às 10:09, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site <https://sei.ufscar.br/autenticacao>, informando o código verificador 1354811 e o código CRC 894A6248.

Agradecimentos

Acredito que não poderia começar os agradecimentos sem agradecer primeiro a Deus pela honra e capacidade de ter concluído não apenas este trabalho, mas também a universidade.

Gostaria de agradecer a minha família, em especial a minha mãe, Miako, que me apoiou em todo tempo e que passou noites em claro comigo. Brincadeira à parte, mas ela está se formando junto comigo.

Agradeço também ao meu orientador, Prof. Dr. Renato Fernandes Cantão. Se tem algo que aprendi é que professores podem mudar nossa visão da faculdade, e que orientadores podem mudar nossa visão de mundo. Agradeço imensamente, não apenas por topar essa ideia maluca que tive, mas também pelas horas de conversas, reuniões, conselhos e risadas. Também gostaria de agradecer ao meu orientador da Iniciação Científica, Prof. Dr. Sérgio Dias Campos, que também topou essa ideia maluca. Ao CNPq pelo apoio financeiro durante a Iniciação Científica.

Aos meus amigos, Gregório, Jullyana e Patrícia que fizeram esse final suportável, por todos os conselhos e conversas que me impediram de desistir. E por fim, aos meus colegas de curso, Daniel e Erika por todo o suporte ao decorrer desses anos.

*“Pela fé entendemos que o universo foi formado pela palavra de Deus,
de modo que o que se vê não foi feito do que é visível.”*

Hebreus 11:3

Resumo

MION, Sara Fernanda de Jesus. A utilização de Redes Neurais Artificiais no estudo da Constante de Acoplamento da Cromodinâmica Quântica. 2024. Trabalho de Conclusão de Curso (Licenciatura Plena em Física) – Universidade Federal de São Carlos, Sorocaba, 2024.

O progresso dos algoritmos computacionais têm proporcionado mais praticidade e conforto para a sociedade. Uma classe de algoritmos que tem se destacado é a de Aprendizado de Máquina (do Inglês, *machine learning*), sendo utilizada no dia-a-dia em mecanismos de buscas, de tradução de textos e até mesmo na análise de plantações de soja. A utilização na área de física de partículas e altas energias ainda é limitada, porém, promissora.

A Cromodinâmica Quântica é a teoria do Modelo padrão que estuda a interação de partículas que possuem cor: os quarks e os glúons. Apesar de ser muito bem consolidada no regime perturbativo, ainda há muito a se estudar no regime não perturbativo.

É a partir dessa perspectiva que no presente trabalho são investigadas maneiras de utilizar algoritmos de *Aprendizado de Máquina*, mais especificamente de redes neurais, para o estudo da constante de acoplamento no regime não perturbativo. O estudo foi realizado com as redes do tipo *Feed Forward* e *Long Short-Term Memory* para diferentes topologias, utilizando as funções de ativação ReLU, sigmoide, tangente hiperbólica e ELU.

Palavras-chave: aprendizado de máquina, redes neurais, aprendizado profundo, cromodinâmica quântica, constante de acoplamento, polo de Landau.

Abstract

Computational algorithms and their advancements have brought increased comfort to society. One standout algorithm is machine learning, utilized not only in everyday tasks such as search mechanisms and text translation but also in the analysis of soy crops. Although its use in the field of particle physics and high-energy physics is currently limited, it holds promise.

Quantum Chromodynamics is the theory of the Standard Model that explores the interaction of color-carrying particles: quarks and gluons. Despite being well-established in the perturbative regime, there is still much to study in the non-perturbative regime.

From this perspective, the present work explores the application of machine learning algorithms, specifically neural networks, to investigate the coupling constant in the non-perturbative regime. Different network architectures, including Feed Forward and Long Short-Term Memory (LSTM), were studied using activation functions such as ReLU, Sigmoid, Hyperbolic Tangent, and ELU.

Keywords: machine learning, neural networks, deep learning, quantum chromodynamics, coupling constant, Landau pole.

Lista de Figuras

Figura 1	– Representação de um neurônio biológico.	28
Figura 2	– Neurônio Artificial	29
Figura 3	– Elementos típicos de uma rede neural. Em verde claro estão representados os neurônios da camada de entrada, em azul camadas ocultas e em vermelho claro a camada de saída.	29
Figura 4	– Algoritmos de Aprendizado de Máquina. Na categoria de Aprendizado Profundo tem-se os algoritmos de <i>Convolutional Neural Network</i> (CNN), <i>Deep Neural Network</i> (DNN), <i>Deep Belief Network</i> (DBF), <i>Recurrent Neural Network</i> (RNN), <i>Bimodal Deep AutoEncoder</i> (BDAE) e <i>Voting Ensembles</i> (vEn), enquanto em Aprendizado Raso tem-se <i>Support Vector Machine</i> (SVM), <i>K-Nearest Neighbors</i> (kNN), <i>Random Forests</i> (RF), <i>Decision Trees</i> (DT), <i>Multi-Layer Perceptron</i> (MLP) e <i>Probabilistic Neural Network</i> (PNN).	30
Figura 5	– Exemplo de uma função de custo $f(x, y) = x^2 + y^2$	32
Figura 6	– Função de ativação ReLU e sua derivada.	34
Figura 7	– Função de ativação sigmoide e sua derivada.	34
Figura 8	– Função de ativação tangente hiperbólica e sua derivada.	35
Figura 9	– Função de ativação ELU e sua derivada.	35
Figura 10	– Fluxo de informações para uma rede neural de topologia <i>Feed Forward</i>	36
Figura 11	– Padrão de conexão dos neurônios em uma rede neural de topologia RNN.	37
Figura 12	– Padrão de conexão dos neurônios em uma rede neural de topologia LSTM.	37
Figura 13	– Partículas elementares do Modelo Padrão.	39
Figura 14	– Caminho óctuplo dos bárions.	41
Figura 15	– Constante de acoplamento da QCD.	45
Figura 16	– Constante de acoplamento $\alpha(Q^2)$ para a região perturbativa com a abscissa normal em (a) e abscissa invertida em (b).	49
Figura 17	– Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta pela função de ativação ReLU, 20 camadas e 50 neurônios.	52
Figura 18	– Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação sigmoide, 5 camadas e 50 neurônios.	52
Figura 19	– Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação tangente hiperbólica, 10 camadas e 50 neurônios.	53

Figura 20 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação ELU, com 50 camadas e 3 neurônios.	54
Figura 21 – Extrapolação de $\alpha(Q^2)$ para a região não perturbativa em (a) e perturbativa em (b) com a topologia composta por função de ativação ELU, 5 camadas e 3 neurônios.	54
Figura 22 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação ELU, 10 camadas e 20 neurônios.	55
Figura 23 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação ELU, 5 camadas e 25 neurônios.	55
Figura 24 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação ELU, 3 camadas e 25 neurônios.	56
Figura 25 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação ELU, 3 camadas e 15 neurônios.	56
Figura 26 – Constante de acoplamento $\alpha(Q^2)$ para a região perturbativa com a abscissa normal em (a) e abscissa invertida em (b).	57
Figura 27 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 5 camadas e 25 neurônios.	58
Figura 28 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 20 camadas e 15 neurônios.	59
Figura 29 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) , com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 50 camadas e 25 neurônios.	59
Figura 30 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 3 camadas e 25 neurônios.	60
Figura 31 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 3 camadas e 20 neurônios.	60
Figura 32 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 5 camadas e 15 neurônios.	61

Figura 33 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 3 camadas e 3 neurônios.	61
Figura 34 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 5 camadas e 3 neurônios.	62
Figura 35 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa e não perturbativa com a topologia composta por função de ativação ReLU, 7 janelas, 2 camadas e 50 neurônios.	65
Figura 36 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa e não perturbativa com a topologia composta por função de ativação ELU, 5 janelas, 2 camadas e 50 neurônios.	65

Lista de Tabelas

Tabela 1	– Relação entre as simetrias e as leis de conservação.	40
Tabela 2	– Número bariônico e estranheza para os quarks up, down, strange e os antiquarks.	42
Tabela 3	– Versões utilizadas de acordo com cada biblioteca utilizada na implementação das redes neurais.	48
Tabela 4	– Descrição dos parâmetros usados para formar as topologias testadas neste trabalho. Foram combinadas três funções de ativação, três quantidades de camadas e quatro quantidades de neurônios por camada.	48
Tabela 5	– Topologias utilizadas para uma rede do tipo LSTM, onde os dados de treinamento foram uma junção das Equações (4.1) e (4.2), para 50 neurônios.	49
Tabela 6	– Descrição dos parâmetros usados para formar as topologias testadas neste trabalho. Foram combinadas três funções de ativação, três quantidades de camadas e quatro quantidades de neurônios por camada.	51
Tabela 7	– Melhores topologias encontradas a partir da métrica MAPE, considerando a Equação (4.3).	52
Tabela 8	– Melhores topologias encontradas a partir da Métrica MAPE para a função de ativação ELU. As equações utilizadas em cada caso constam da terceira coluna.	53
Tabela 9	– Melhores topologias encontradas a partir da Métrica MAPE para a função de ativação ELU. As equações utilizadas em cada caso constam da terceira coluna.	58
Tabela 10	– Comparativo entre as melhores métricas obtidas para o treinamento normal e para o treinamento com a abscissa invertida, considerando as Equações (4.1), (4.2) e (4.3), para a função de ativação ELU.	63
Tabela 11	– Evolução da métrica MAPE, para diversos intervalos, considerando a extrapolação no regime perturbativo e a Equação (4.1).	63
Tabela 12	– Relação entre a entrada e a saída com janelas de tamanho J em uma série temporal.	64
Tabela 13	– Topologias utilizadas para uma rede do tipo LSTM, onde os dados de treinamento foram uma junção das Equações (4.1) e (4.2), para 50 neurônios.	64
Tabela 14	– Melhores topologias encontradas a partir da Métrica MAPE, considerando a Equação (4.3), para 50 neurônios	65

Sumário

1	INTRODUÇÃO	21
2	APRENDIZADO DE MÁQUINA	23
2.1	Treinamento e Teste	24
2.2	Tipos de Aprendizado	25
2.2.1	Aprendizado Supervisionado	25
2.2.1.1	Regressão	25
2.2.1.2	Classificação	26
2.2.2	Aprendizado Não Supervisionado	26
2.2.2.1	Agrupamento	27
2.2.3	Aprendizado Semi-Supervisionado	27
2.2.4	Aprendizado Por Reforço	27
2.3	Redes Neurais	28
2.3.1	Topologias	31
2.3.2	Métricas de Avaliação e Função de Custo	31
2.3.2.1	Algoritmos de Otimização	32
2.3.3	Funções de Ativação	33
2.3.3.1	Unidade Linear Retificada – ReLU	33
2.3.3.2	Sigmoide	33
2.3.3.3	Tangente Hiperbólica – Tanh	33
2.3.3.4	Unidade Exponencial Linear – ELU	34
2.3.4	Topologias típicas	36
2.3.4.1	<i>Feed Forward</i> – FF	36
2.3.4.2	<i>Recurrent Neural Network</i> – RNN	36
2.3.4.2.1	<i>Long Short Term Memory</i> – LSTM	36
3	FÍSICA DE PARTÍCULAS	39
3.1	Modelo Padrão – Uma visão Geral	39
3.1.1	Os Quarks	40
3.1.2	Cromodinâmica Quântica	43
4	METODOLOGIA	47
4.1	Geração dos dados	47
4.2	Implementação computacional	47
4.3	Análise de Dados	49

5	RESULTADOS E DISCUSSÃO	51
5.1	<i>Feed Forward</i>	51
5.2	LSTM	64
6	CONCLUSÕES	67
	Referências	69

1 Introdução

Em Ciência da Computação, o estudo e a pesquisa dos algoritmos de Aprendizado de Máquina têm se intensificado nas últimas décadas, particularmente devido ao advento da Internet, o que possibilitou um aumento exponencial na quantidade de dados coletados. A função destes algoritmos é a de identificar padrões para, a partir destes, serem realizadas previsões a partir de dados inéditos (GÉRON, 2019).

Dentre os inúmeros exemplos da utilização desses algoritmos no cotidiano, pode-se citar: recomendações de filmes e séries online (LADEIRA, 2019); assistentes virtuais como a Cortana (Microsoft), Siri (Apple) e Alexa (Amazon) (SGARBOSA; VECHIO, 2020); na saúde, na classificação de lesões dermatológicas benignas e malignas, na prevenção de eventos cardiovasculares ou na análise de critérios para identificação de pessoas pertencentes ao transtorno do espectro autista (PAIXÃO et al., 2022).

A Física de Partículas é uma área da Física que estuda os constituintes da matéria, identificando-as como partículas e mediadoras, tendo como base as Teorias Quânticas de Campos, podendo ser separada em:

Eletrodinâmica Quântica (QED) responsável por estudar a interação entre partículas e mediadores que interagem por meio da carga elétrica.

Teoria Fraca responsável por estudar os decaimentos de partículas, assim como a emissão de uma partícula conhecida como neutrino. Essa teoria é que guia o estudo da força nuclear fraca.

Cromodinâmica Quântica (QCD) responsável por estudar a interação entre partículas e mediadores que interagem por meio da carga conhecida como cor. Essa teoria é que guia o estudo da força nuclear forte (THOMSON, 2013; GRIFFITHS, 1987).

O presente trabalho tem seu foco nesta última, a Cromodinâmica Quântica.

Apesar da grande utilização de Redes Neurais Artificiais (RNA) no cotidiano, na Física de altas energias seu uso tem sido restrito a alguns grupos de pesquisa. Alguns exemplos que são possíveis citar: a utilização de *autoencoders*, algoritmo que possui como característica a cópia dos dados de entrada nos dados de saída para minimização da função de erro, na procura de partículas exóticas realizadas por Baldi, Sadowski e Whiteson (2014); a utilização de RNA do tipo *convolutional neural network*, um tipo de algoritmos que aplica um ‘filtro’ nos dados de entrada e reduz as dimensões desses dados, por Baldi, Bauer et al. (2016) para a classificação de partículas presentes em um jato de partículas, utilizando a simulação de Monte Carlo para a geração dos dados. Utiliza-se também os algoritmos de regressão paramétrica, algoritmo que determina como as variáveis independentes interferem na variável dependente, nos estudos das teorias de calibre, como aponta Shanahan, Trewartha e Detmold (2018).

Já na QCD, restringe-se ainda mais o uso das RNA, sendo utilizados na QCD holográfica (HASHIMOTO et al., 2018; AKUTAGAWA; HASHIMOTO; SUMIMOTO, 2020), e na transição de fase da QCD (PANG et al., 2019). Portanto, este trabalho explora essa área, para verificar se é possível a utilização das RNA no estudo da constante de acoplamento da QCD.

O trabalho segue organizado da seguinte forma: a Seção 2 consiste no conteúdo referente ao Aprendizado de Máquina, focando em Aprendizado Profundo. A Seção 3 traz as informações referentes à Física de Partículas, com ênfase no desenvolvimento do modelo a quarks e na Cromodinâmica Quântica. A Seção 4 contém a metodologia utilizada, definindo as topologias utilizadas no presente trabalho. A Seção 5 contém os resultados, assim como as discussões pertinentes aos resultados encontrados, e, por fim, na Seção 6 encontra-se a conclusão.

2 Aprendizado de Máquina

O Aprendizado de Máquina (em inglês, *Machine Learning*) teve Arthur Lee Samuel como um de seus pioneiros. Na década de 1950 ele desenvolveu um algoritmo capaz de aprender jogadas de damas. Para ele, o Aprendizado de Máquina (AM) consistia na identificação de padrões quando o algoritmo não era especificamente programado para isso, sendo que o jogo de damas era um ótimo recurso para desenvolver o AM, pois fornecia algumas características importantes para estes estudos, sendo estas:

1. Não pode ser determinístico;
2. Um objetivo final deve existir;
3. As regras precisam ser muito bem conhecidas;
4. Os processos de aprendizagem precisam ser testados;
5. A atividade precisa ser de conhecimento geral, pois o programa poderá ser utilizado por mais pessoas (SAMUEL, 1959).

É possível definir o AM como um conjunto de técnicas e tecnologias responsáveis por determinar como um computador “aprende”, ou seja, como programar a capacidade de identificar os parâmetros de um problema a fim de generalizá-lo (MELLO; PONTI, 2018). Para isso, é necessário haver a interdisciplinaridade entre Ciência da Computação e Matemática, pois além da programação, utiliza-se de métodos estatísticos para realizar a generalização (MÜLLER; GUIDO, 2016).

Os exemplos onde o AM estão presentes são diversos: possíveis faltas de candidatos em consultas marcadas no Sistema Único de Saúde (BERTOZZO, 2019), a variação de valores de ações na bolsa de valores (SANTOS, 2020), e a utilização para o reconhecimento de fala a partir da transcrição automática de fonemas (DIJKSTRA, 2021). Outro exemplo que pode-se citar é o de reconhecimento de números escritos a mão, com o intuito de digitalizá-los. Para isso, é necessário que o programa realize a leitura dos escritos, de forma que seja possível convertê-los em dados computacionais (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Como toda tecnologia, o AM tem suas limitações como a restrição na generalização, o aperfeiçoamento constante na implementação dos algoritmos, além da busca por novos algoritmos com melhor desempenho e a quantidade de dados requerida para seu uso efetivo (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Os dados podem ser categorizados em dados de entrada e dados de saída. Seja $X = (X_1, X_2, \dots, X_N)$ um conjunto de dados denominado *dados de entrada*, tal que $X_i \in \mathbb{R}^N$, associado a um *conjunto de saída* $y = (y_1, y_2, \dots, y_N)$, sendo N a quantidade de dados fornecida. Dados esses conjuntos de

entrada e saída, estes serão divididos em dois subconjuntos denominados de *conjunto de treinamento* e *conjunto de teste*. O primeiro é fornecido ao algoritmo para que ele consiga extrair padrões presentes, assim como realizar a associação entre a entrada X e a saída y . Já o segundo é utilizado na verificação do desempenho do algoritmo após esse treinamento. Comumente, o conjunto de treinamento corresponde a 70% dos dados, enquanto o conjunto de teste corresponde a 30% dos dados, mas essa proporção pode ser alterada dependendo da questão a ser abordada (MÜLLER; GUIDO, 2016; GÉRON, 2019).

Nas próximas seções serão apresentados alguns conceitos básicos de AM.

2.1 Treinamento e Teste

Para usar de forma efetiva um algoritmo de AM são necessários alguns passos essenciais e comuns à maioria das aplicações, listados a seguir:

Aquisição dos dados de treinamento e de teste Na utilização de algoritmos de AM é necessária uma grande quantidade de dados para o treinamento e testes. Com o advento da internet, a quantidade de dados produzidos diariamente aumentou exponencialmente, o que se tornou uma grande vantagem para a utilização de algoritmos de AM (GÉRON, 2019).

Limpeza e tratamento dos dados A limpeza de dados consiste na verificação de quais informações presentes no conjunto serão relevantes para o problema, aferir se esses dados estão no formato correto, ou se existem dados faltantes ou simplesmente errôneos.

Como exemplo, pode-se supor que eles sejam uma tabela: será necessário determinar quais colunas são importantes para as informações que se desejam estudar. Logo, faz sentido selecionar colunas específicas necessárias para aquele problema. É importante lembrar que a utilização de uma grande quantidade de dados, implica em muita memória utilizada. Logo, esse processo de limpeza dos dados, garante uma maior fluidez no algoritmo.

Treinamento Para que um determinado algoritmo tente encontrar padrões, é necessário treiná-lo, utilizando os dados de treinamento. O tipo de aprendizagem, e portanto as características intrínsecas do treinamento, está relacionado com as características dos dados (MÜLLER; GUIDO, 2016).

Teste Um bom programa de AM é aquele que permite a generalização. Depois que o modelo é treinado, utiliza-se os dados de teste para verificar o desempenho do algoritmo selecionado, identificando sua capacidade de generalização a partir de resultados conhecidos (MÜLLER; GUIDO, 2016).

2.2 Tipos de Aprendizado

É possível categorizar os algoritmos conforme o tipo de aprendizado: supervisionado, não supervisionado, semi-supervisionado e por reforço.

2.2.1 Aprendizado Supervisionado

O Aprendizado Supervisionado (AS), consiste na utilização de *rótulos* nos dados de entrada, que em geral corresponde ao conjunto de saída, de forma que a partir de dados inéditos, seja possível realizar previsões. Como já definido anteriormente, dado um conjunto de entradas na forma $X^T = (X_1, \dots, X_N)$, tal que $X_i \in \mathbb{R}^N$, o objetivo de um algoritmo de AS é prever as saídas $y = (y_1, \dots, y_N)$. Para isso, considera-se que (X, y) são variáveis aleatórias e a densidade de probabilidade conjunta seja $\Pr(X, y)$, ou seja, a densidade de probabilidade para um espaço de duas ou mais dimensões, caracterizando, assim a aprendizagem supervisionada como a estimativa do cálculo da densidade da probabilidade condicional $\Pr(y|x)$ (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Dentro do AS, existem duas classes de algoritmos que se destacam: a *regressão* e a *classificação*.

2.2.1.1 Regressão

A *regressão* consiste na aproximação dos dados com uma função polinomial. Supondo que a melhor forma de aproximar os dados seja a partir de uma reta, é possível utilizar a regressão linear. Dessa forma, a reta será semelhante à Equação (2.1), para N dimensões.

$$\hat{y} = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \dots + \theta_N X_N, \quad (2.1)$$

sendo $\theta_i \in \mathbb{R}^N$ os coeficientes da regressão e $\hat{y} \in \mathbb{R}^N$ o resultado da regressão.

Quando limitado a $N = 1$, (2.1) fica resumida a $\hat{y} = \theta_0 + \theta_1 X_1$; logo, a função do algoritmo será a de determinar os parâmetros θ_0 e θ_1 que melhor explicam os dados fornecidos.

Se os dados possuem um caráter não-linear, utiliza-se a regressão polinomial. Para N dimensões, têm-se

$$\hat{y} = \theta_0 + \theta_1 X_1 + \theta_2 X_2^2 + \theta_3 X_3^3 + \dots + \theta_N X_N^N. \quad (2.2)$$

Como no exemplo da regressão linear, é possível citar a predição em valores de habitações em uma determinada região. Quando um algoritmo de regressão usa como dados de treinamento as informações de imóveis de uma região, assim como os seus valores, o modelo consegue prever qual seria o valor de uma casa com determinadas características na mesma região. Logo, ele retorna um valor numérico, uma característica da regressão (GÉRON, 2019).

A *regressão logística*, também conhecida como classificação logística, consiste na determinação da probabilidade de determinado evento ocorrer ou não. Apesar de ser chamada de regressão, ela pode ser considerada um híbrido entre classificação e regressão. É determinada pela Equação (2.3)

$$P(\hat{y} = 1) = \frac{1}{1 + e^{-g(X)}}, \quad (2.3)$$

onde $g(X) = \theta_0 + \theta_1 X_1 + \dots + \theta_N X_N$. Logo, a regressão consiste em determinar os valores dos coeficientes $\theta_i, i = 0, \dots, N$, e a classificação consiste na determinação das probabilidades de acontecer determinado evento.

Como exemplo, é possível citar estudos de [Montevechi et al. \(2022\)](#) que utilizaram a classificação logística para determinar a probabilidade de risco ao oferecer crédito para alguém.

Como a regressão logística possui um caráter classificatório, alguns autores a consideram um algoritmo de classificação, que será visto a seguir.

2.2.1.2 Classificação

Como a *classificação* é uma classe de algoritmo de AS, é necessário que os dados de entrada e de saída sejam rotulados. Apesar de ser similar à regressão, a sua diferença está no tipo de dado que ela retorna: enquanto a regressão retorna um dado quantitativo, a classificação retorna um dado qualitativo ([GÉRON, 2019](#)).

Considerando um espaço no qual o conjunto de dados esteja dividido em diferentes classes $K = K_1, K_2, \dots, K_K$ disjuntas ($K_i \cap K_j = \emptyset$), para o K -ésimo indicador, o ajuste linear é dado por

$$\hat{y}_K = \theta_{K0} + \theta_K^T X,$$

tal que, similar ao caso da regressão, θ são os coeficientes da interpolação de dimensão K , e X são as entradas em sua forma vetorial.

Já, o hiperplano que separa a classe K_i da classe K_j pode ser dado por $\hat{y}_{K_i}(X) = \hat{y}_{K_j}(X)$, tal que $\{X : (\theta_{K0} - \theta_{K_j0}) + (\theta_{K_i} - \theta_{K_j})^T X = 0\}$. O objetivo de um algoritmo de classificação é encontrar os coeficientes que ajuste o hiperplano ([HASTIE; TIBSHIRANI; FRIEDMAN, 2009](#)).

Como exemplo, é possível citar uma pesquisa que utilizou um algoritmo de classificação para tentar determinar o desempenho de estudantes ao longo do primeiro semestre de Engenharia em uma universidade da Índia. As entradas consistiam em informações dos alunos, como nome, classe social, notas nas disciplinas de exatas, além de notas dos exames de admissão. A acurácia de tal algoritmo foi em torno de 75% ([ADHATRAO et al., 2013](#)).

2.2.2 Aprendizado Não Supervisionado

Diferente do Aprendizado Supervisionado, o Aprendizado Não Supervisionado (ANS), não possui rótulos em seus dados. De maneira geral, o objetivo de um algoritmo de ANS é detectar padrões no conjunto de dados de entrada ([GÉRON, 2019](#)).

Em outras palavras, partindo de um conjunto com N observações $X^T = (X_1, X_2, \dots, X_N)$, o objetivo de um algoritmo de ANS é determinar sem a ajuda de um supervisor a densidade de probabilidade $\Pr(X)$ ([HASTIE; TIBSHIRANI; FRIEDMAN, 2009](#)). O principal algoritmo de ANS é o de *agrupamento*.

2.2.2.1 Agrupamento

O *agrupamento* consiste na organização de dados, de forma que esses caracterizem os dados que estão representando, ou seja, consiste na organização em classes, tal que os elementos de uma mesma classe possuam particularidades em comum. Algoritmos de agrupamento são um tipo de algoritmo de classificação que depende apenas dos dados de entrada para o seu funcionamento. Os dados de entrada são chamados de matriz de proximidade, que são sempre simétricas (JAIN; DUBES, 1988).

Os algoritmos de ANS são amplamente utilizados para detecção de anomalias e padrões não esperados e, dadas suas características, foram utilizados durante a pandemia de COVID-19 para análise do avanço do Coronavírus, assim como na predição do avanço da pandemia (MUHTASIM; MASUD, 2023).

2.2.3 Aprendizado Semi-Supervisionado

O aprendizado Semi-Supervisionado consiste em uma junção dos dois tipos apresentados anteriormente. Isso significa que parte dos dados são rotulados, enquanto a outra parte permanece sem os rótulos (FONTANA, 2020). Os dados rotulados são utilizados como um auxílio para os dados não rotulados, de forma que possam ser usados tanto para a classificação, quanto para o agrupamento (BRUCE, 2002). Uma possibilidade do Aprendizado Semi-Supervisionado, é a utilização dos dados rotulados para encontrar os preditores fracos, e em seguida, utilizar esses preditores para inicializar os dados não rotulados (BLUM; MITCHELL, 1998).

Como exemplo, é possível citar os estudos com a soja. Como o Brasil é um grande produtor de soja, a preocupação com sua qualidade é constante. É possível elaborar um algoritmo de Aprendizado Semi-Supervisionado para a detecção de doenças e de plantas invasoras da soja. A partir de imagens das plantas, é possível analisá-las com base nos parâmetros de cor, gradiente, textura e forma para identificar as possíveis mazelas que a plantação esteja sofrendo (TETILA, 2019).

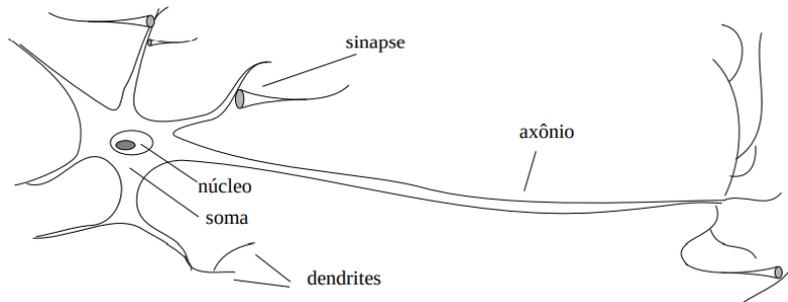
2.2.4 Aprendizado Por Reforço

O Aprendizado por Reforço consiste no aprendizado quando há a interação com o ambiente. Comumente utilizado na robótica, é empregado de forma que a máquina consiga tomar uma decisão a partir da recompensa ou da punição que ela receberá. Parte-se da premissa de que uma recompensa sempre estará reforçando a última decisão tomada. No caso da técnica de Diferenças Temporais, esse reforço precisa ser propagado no tempo, fazendo, assim, com que o robô aprenda (GARIA; ROMERO, 2002).

2.3 Redes Neurais

Os neurônios são células do sistema nervoso que possuem como função a passagem de impulsos elétricos. São compostos por axônios, corpo celular, dendritos, e sinapses como pode ser visto na Figura 1. A sinapse é a parte responsável por transmitir esses impulsos elétricos aos neurônios seguintes (RAUBER, 2014).

Figura 1 – Representação de um neurônio biológico.



Fonte: Retirado de Rauber (2014).

As redes neurais biológicas são conjuntos de neurônios interligados entre si, visando transmitir esses impulsos. As Redes Neurais Artificiais (em Inglês, *Artificial Neural Networks*) são algoritmos computacionais baseados nas redes neurais biológicas, compostas por neurônios que têm o papel de aproximadores de funções (HORNIK; STINCHCOMBE; WHITE, 1989). Um neurônio artificial pode ser verificado na Figura 2. Nas próximas seções, serão apresentados os principais elementos que constituem o AP, sendo eles:

Denomina-se de *camada* um conjunto de neurônios que realizam operações específicas de tal modo que há a organização e a estruturação do fluxo de informações. De maneira geral, um neurônio de uma camada $M - 1$ está conectado a todos os neurônios da camada M . Em redes neurais mais simples, existem três tipos de camadas:

Camada de entrada: trata-se da primeira camada, a que recebe os dados.

Camadas ocultas: localizadas entre a camada de entrada e de saída.

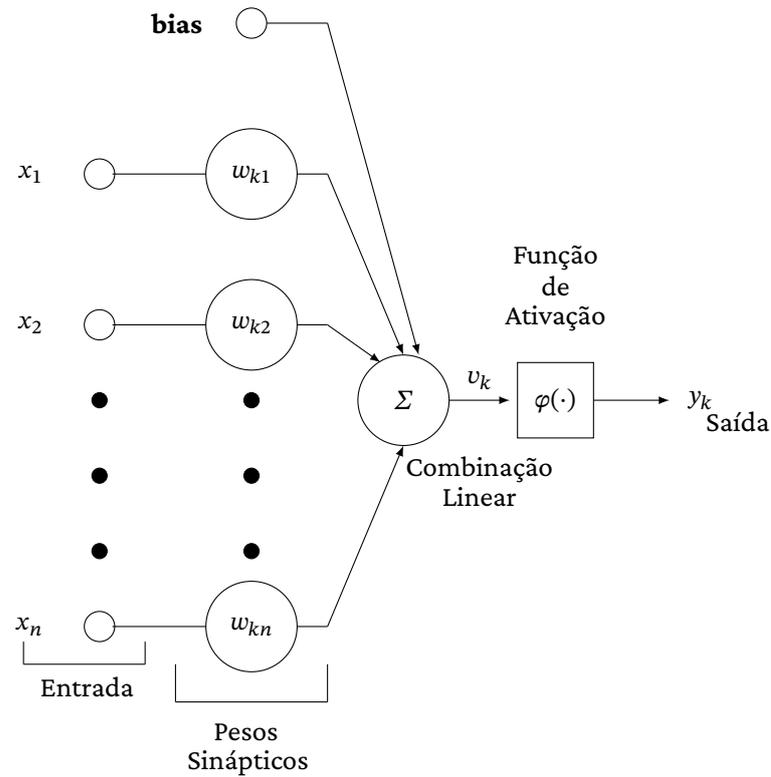
Camada de saída: última camada da rede, produzindo o resultado final da rede neural (AGGARWAL, 2018).

Estes elementos podem ser verificados na Figura 3.

Na conexão entre dois neurônios k e j existe um peso $w_{kj} \in \mathbb{R}$, atualizado a partir de um *otimizador*. A quantidade de vezes que os dados são processados pela rede neural, é chamado de *épocas*.

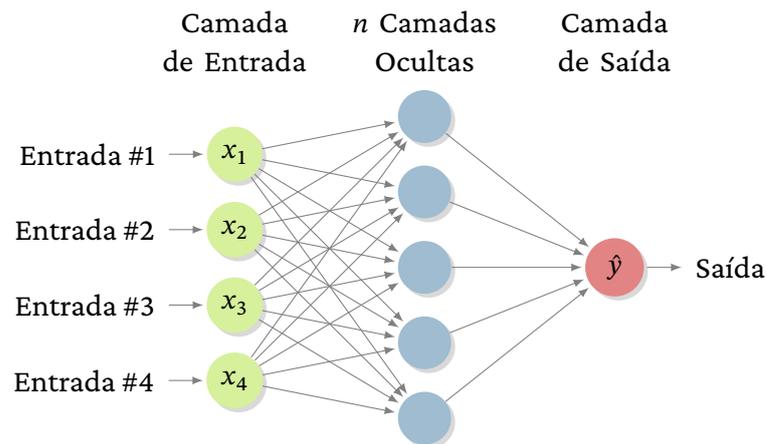
Do ponto de vista da quantidade de camadas, um algoritmo de AM, pode ser dividido em duas categorias: Aprendizado Profundo (AP) (em Inglês *deep learning*), ou Aprendizado Raso (AR) (em Inglês, *shallow learning*). De maneira geral, o AP utiliza uma grande quantidade de camadas,

Figura 2 – Neurônio Artificial



Fonte: Retirado de [André Ricardo Gonçalves \(2018\)](#).

Figura 3 – Elementos típicos de uma rede neural. Em verde claro estão representados os neurônios da camada de entrada, em azul camadas ocultas e em vermelho claro a camada de saída.



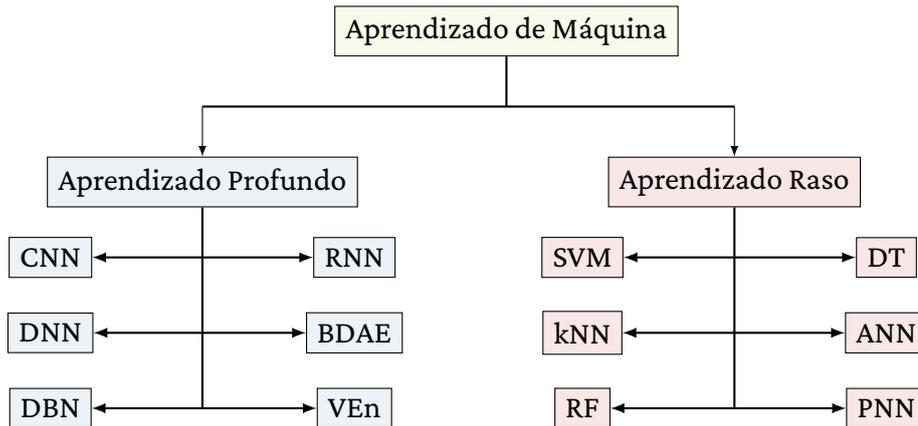
Fonte: Adaptado de [Shiruru \(2015\)](#).

enquanto o AR se limita a uma ou duas camadas. A utilização de cada uma das técnicas depende do problema em questão, lembrando que quanto maior o número de camadas e neurônios, maior será o consumo computacional e o tempo para a análise dos modelos ([DELANCEY et al., 2020](#)).

Este trabalho irá se ater aos algoritmos de AP.

Neste contexto, é possível separar alguns algoritmos de AM conforme a Figura 4.

Figura 4 – Algoritmos de Aprendizado de Máquina. Na categoria de Aprendizado Profundo tem-se os algoritmos de *Convolutional Neural Network* (CNN), *Deep Neural Network* (DNN), *Deep Belief Network* (DBN), *Recurrent Neural Network* (RNN), *Bimodal Deep AutoEncoder* (BDAE) e *Voting Ensembles* (VEn), enquanto em Aprendizado Raso tem-se *Support Vector Machine* (SVM), *K-Nearest Neighbors* (kNN), *Random Forests* (RF), *Decision Trees* (DT), *Multi-Layer Perceptron* (MLP) e *Probabilistic Neural Network* (PNN).



Fonte: Adaptado de Islam et al. (2021).

Uma Rede Neural Artificial (RNA) típica possui como elementos básicos, além do número de camadas (e respectivos neurônios) os seguintes elementos, que serão abordados ao longo da Seção:

- Funções de ativação.
- Topologias típicas.
- Métricas de avaliação.
- Algoritmos de otimização.

Para apresentação dos parâmetros de uma RNA, será introduzida a seguinte questão: seria possível relacionar entradas e saídas de funções, de maneira que seus valores possam ser extrapolados? Para respondê-la, considere um vetor $X \in \mathbb{R}^n$ que contém os dados de entrada, e o vetor $v \in \mathbb{R}^n$ que contém os dados de saída de um neurônio. Inicialmente, o neurônio era uma combinação linear, que está representado pela Equação (2.4).

$$v_k = \sum_{j=1}^m w_{kj} x_j, \quad (2.4)$$

w_{kj} são os pesos e x_j as entradas. Considerando o neurônio como apenas uma combinação linear entre as entradas e os pesos, não é possível classificar, ou até mesmo realizar uma regressão em

um problema que não seja linear. Sendo assim, é necessário acrescentar uma não-linearidade no neurônio, papel da função de ativação (DOMINGOS, 2012).

Funções de ativação, são responsáveis por introduzir não-linearidades em uma rede neural, como é possível verificar na Equação (2.5) (HAYKIN, 2001)

$$\hat{y}_k = \varphi(v_k + b_k), \quad (2.5)$$

\hat{y}_k é a saída do neurônio, φ é a função de ativação e b_k é o *bias* do neurônio. O *bias* é um parâmetro de cada neurônio, que permite que a rede neural aprenda de forma mais flexível, incorporando a parte invariante do neurônio (AGGARWAL, 2018). Essas não-linearidades permitem uma flexibilidade maior na aproximação de funções.

2.3.1 Topologias

É possível categorizar uma rede neural conforme a sua topologia, ou seja, como neurônios e camadas se conectam entre si. Algumas topologias merecem destaque e serão abordadas mais à frente, como por exemplo *Feed Forward* (FF), *Recurrent Neural Network* (RNN) e *Long Short Term Memory* (LSTM).

2.3.2 Métricas de Avaliação e Função de Custo

Para determinar o desempenho de um modelo de AM, é necessária a utilização de métricas de avaliação. Nessa seção, serão apresentadas as mais utilizadas na prática.

Erro Quadrático Médio, do inglês *Mean Squared Error - MSE*, é dado por

$$\text{MSE} = \frac{1}{N} \sum_{i=0}^N (y_i - \hat{y}_i)^2.$$

Erro Absoluto Médio, do inglês *Mean Absolute Error - MAE*, é dado por

$$\text{MAE} = \frac{1}{N} \sum_{i=0}^N \|y_i - \hat{y}_i\|.$$

Erro Percentual Absoluto Médio, do inglês *Mean Absolute Percentual Error - MAPE*, é dado por

$$\text{MAPE} = \frac{1}{N} \sum_{i=0}^N \left\| \frac{y_i - \hat{y}_i}{y_i} \right\| \times 100\%.$$

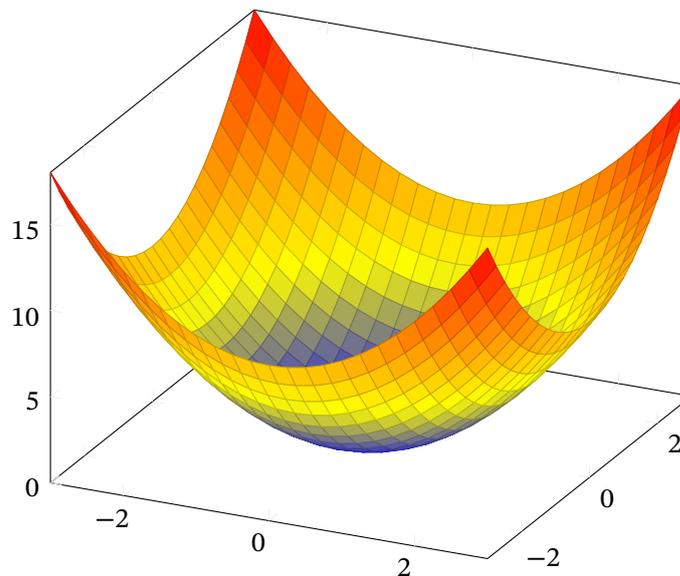
Todas as métricas apresentadas têm uma característica em comum: quanto maior for o seu valor, pior é o desempenho do modelo, contudo as métricas MAE e MSE são mais sensíveis quando existem pontos denominados *outliers*, sendo aqueles que são destoantes do conjunto em questão (CARMO; MELO SILVA, 2023).

Essas métricas são geralmente utilizadas no processo de treinamento da rede, com o intuito de averiguar a *função de custo*, que determina o quão bem o modelo aproxima os dados (GÉRON, 2019).

2.3.2.1 Algoritmos de Otimização

Os algoritmos de otimização utilizam-se do valor fornecido pela *função de custo*, buscando minimizá-la. Para exemplificar o processo de aprendizagem, imagine uma função de custo que possua uma superfície como mostra a Figura 5.

Figura 5 – Exemplo de uma função de custo $f(x, y) = x^2 + y^2$.



Fonte: A autora.

É possível verificar que ela terá um mínimo, ou seja, um ponto onde a função de custo atingirá seu menor valor. Vários algoritmos podem ser usados para encontrar esse ponto de mínimo, sendo um dos mais utilizados o *gradiente descendente*. De forma geral, quando calcula-se o gradiente de uma função, ele aponta para a direção de máximo crescimento. Logo, para encontrar o mínimo, é possível seguir o caminho oposto, por isso gradiente descendente (GÉRON, 2019; HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

De acordo com Géron (2019), a *taxa de aprendizado* determina o tamanho do passo na direção do gradiente descendente, a fim de encontrar o mínimo. Se a taxa de aprendizado for muito alta, há a possibilidade de o algoritmo entrar num ciclo, de forma a nunca chegar a um mínimo. Se a taxa de aprendizado for muito pequena, há a possibilidade do algoritmo demorar muito tempo para convergir. Logo, o valor da taxa de aprendizado é um hiperparâmetro muito importante para a minimização dos erros.

Se a função de custo, por exemplo, for a MSE, o gradiente descendente será calculado a partir da Equação (2.6)

$$\frac{\partial \text{MSE}(\hat{y})}{\partial \hat{y}_i} = \frac{\partial}{\partial \hat{y}_i} \left(\frac{1}{N} \sum_{i=0}^N (y_i - \hat{y}_i)^2 \right) = -\frac{2}{N} \sum_{i=0}^N (y_i - \hat{y}_i). \quad (2.6)$$

O problema dessa forma, também chamada de Gradiente Descendente em Lote, é que ela utiliza todo o conjunto de dados para o cálculo, o que pode levar muito tempo, a depender do tamanho do

problema. Como alternativa, o Gradiente Descendente Estocástico utiliza uma parcela aleatória dos dados para o cálculo do gradiente descendente, diminuindo o tempo de execução. Devido à aleatoriedade desse algoritmo, ele consegue escapar de mínimos locais, porém pode nunca atingir o valor ótimo do mínimo global (GÉRON, 2019).

É possível classificar os otimizadores em duas classes: *forward propagation* (do Inglês, “propagação para frente”), e *backpropagation* (do Inglês, “propagação para trás”). A diferença entre as duas é o sentido de propagação dos pesos. No caso da rede utilizar *forward propagation*, esse sentido ocorre apenas da entrada para a saída, já no caso de *backpropagation*, ocorre em ambos os sentidos (GÉRON, 2019; HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

2.3.3 Funções de Ativação

Alguns exemplos de *funções de ativação* são apresentados a seguir. Como os pesos calculados são através da retropropagação, processo relacionado com a derivada dos valores de saída de cada neurônio, é interessante verificar as derivadas das funções de ativação, para averiguar seu comportamento. A presença da não-linearidade na rede é importante, pois a maioria das questões tratadas com inteligência artificial são não-lineares (GÉRON, 2019).

2.3.3.1 Unidade Linear Retificada – ReLU

A *ReLU*, Equação (2.7), é a mais popular dentre as funções de ativação utilizadas atualmente, por possuir características lineares, permitindo que métodos de minimização de erros sejam mais efetivos, mais precisamente os métodos de gradiente descendente (RAJANAND; SINGH, 2023).

$$\varphi(v) = \begin{cases} v, & \text{se } v \geq 0 \\ 0, & \text{se } v < 0 \end{cases} \quad \varphi'(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ 0, & \text{se } v < 0 \end{cases}. \quad (2.7)$$

A Figura 6 mostra a Função de Ativação ReLU, assim como a sua derivada.

2.3.3.2 Sigmoides

Outra função de ativação bastante utilizada é a *sigmoide*. A característica dela é que seu mínimo é 0 e seu máximo é 1, além de seu gradiente tender a zero para redes neurais mais complexas, de forma que pode dificultar o cálculo correto dos pesos. A Equação (2.8) caracteriza a sigmoide e a sua derivada, que podem ser verificadas na Figura 7.

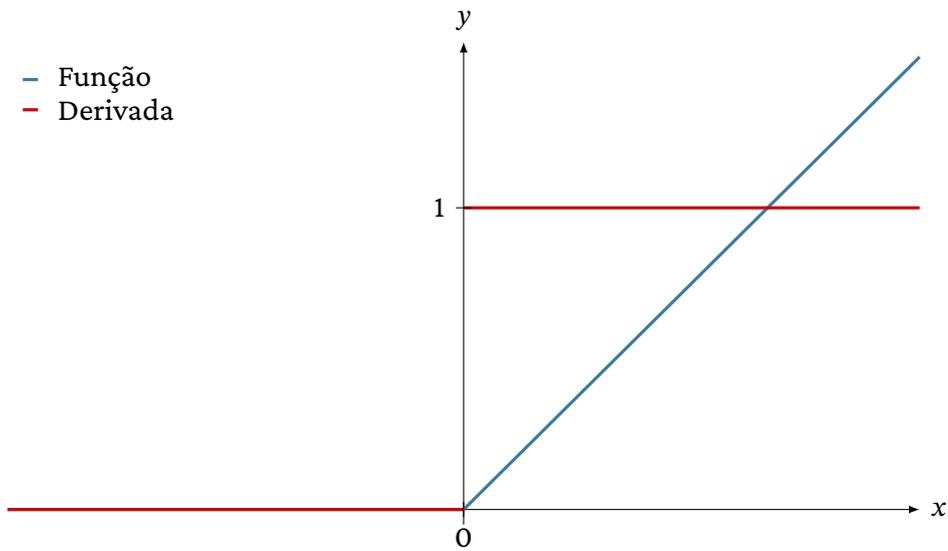
$$\varphi(v) = \frac{1}{1 + e^{-v}} \quad \varphi'(v) = \frac{e^{-v}}{(1 + e^{-v})^2}. \quad (2.8)$$

2.3.3.3 Tangente Hiperbólica – Tanh

A tangente hiperbólica possui características semelhantes às da sigmoide, porém seu máximo é 1 e seu mínimo é -1 . É definida por:

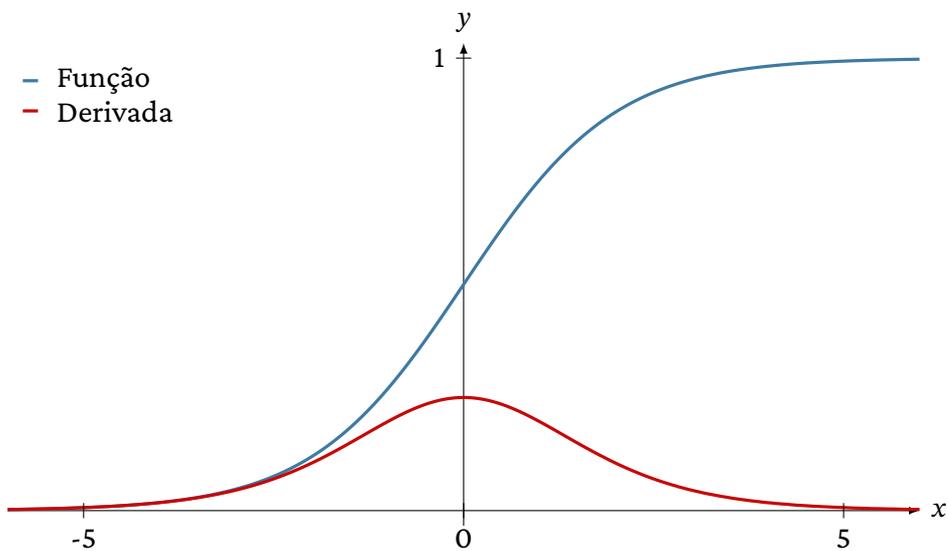
$$\varphi(v) = \frac{e^{2v} - 1}{e^{2v} + 1} \quad \varphi'(v) = 1 - \tanh^2(v). \quad (2.9)$$

Figura 6 – Função de ativação ReLU e sua derivada.



Fonte: Adaptado de [Rajanand e Singh \(2023\)](#).

Figura 7 – Função de ativação sigmoide e sua derivada.



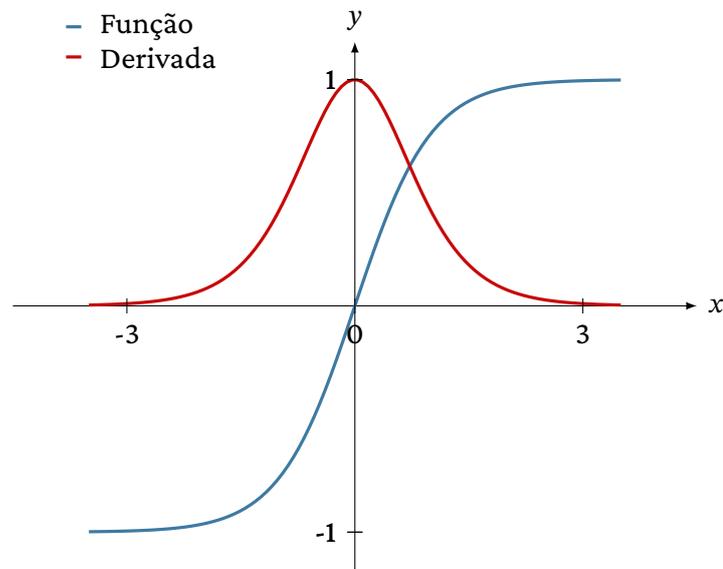
Fonte: Adaptado de [Rajanand e Singh \(2023\)](#).

O gráfico da Equação (2.9) pode ser verificado na Figura 8.

2.3.3.4 Unidade Exponencial Linear – ELU

A principal característica da *ELU* é preservar a identidade das entradas positivas. É descrita pela Equação (2.10), assim como sua derivada, sendo representada graficamente como mostra a

Figura 8 – Função de ativação tangente hiperbólica e sua derivada.

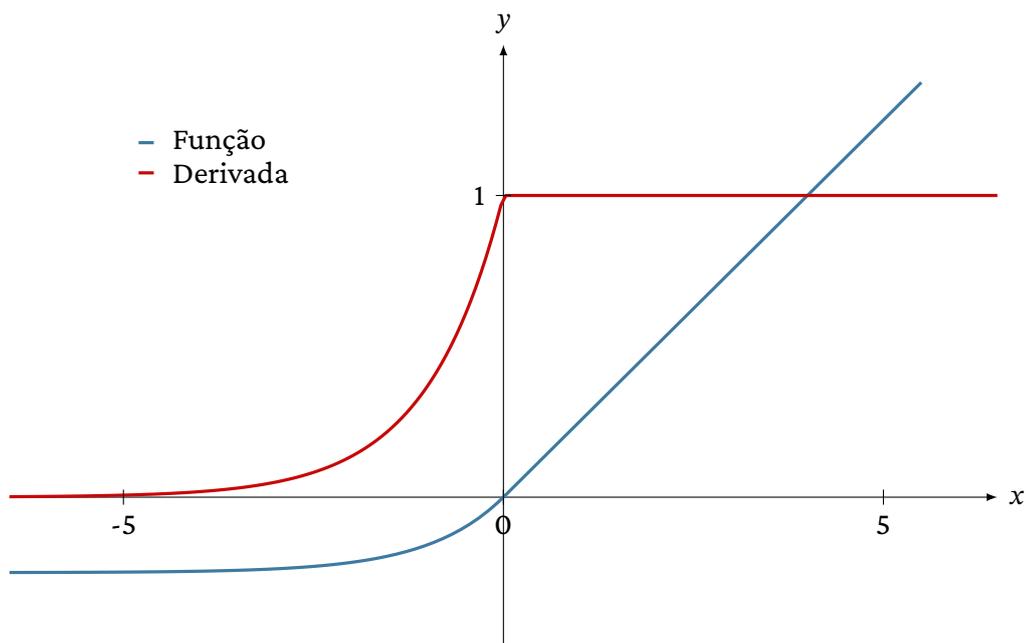


Fonte: Adaptado de [Rajanand e Singh \(2023\)](#).

Figura 9.

$$\varphi(v) = \begin{cases} v, & \text{se } v \geq 0 \\ \alpha(e^v - 1), & \text{se } v < 0 \end{cases} \quad \varphi'(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ \alpha e^v, & \text{se } v < 0 \end{cases}. \quad (2.10)$$

Figura 9 – Função de ativação ELU e sua derivada.



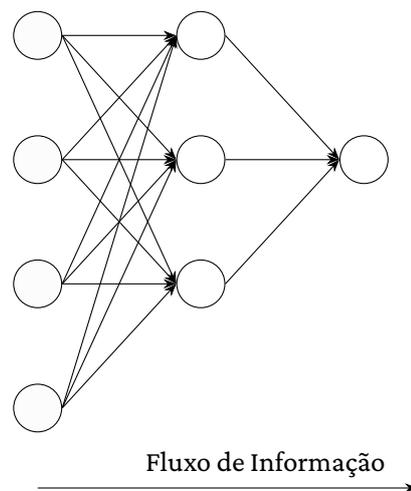
Fonte: Adaptado de [Rajanand e Singh \(2023\)](#).

2.3.4 Topologias típicas

2.3.4.1 *Feed Forward* – FF

Também chamada de perceptron multicamadas, a rede conhecida como *Feed Forward* possui as camadas totalmente conectadas, com um fluxo unidirecional de informações. É caracterizada por possuir camada de entrada, camadas ocultas, camada de saída, pesos em cada neurônio, funções de ativação e, geralmente, utiliza-se da retropropagação, como é possível verificar na Figura 10.

Figura 10 – Fluxo de informações para uma rede neural de topologia *Feed Forward*.



Fonte: Adaptado de André Ricardo Gonçalves (2018).

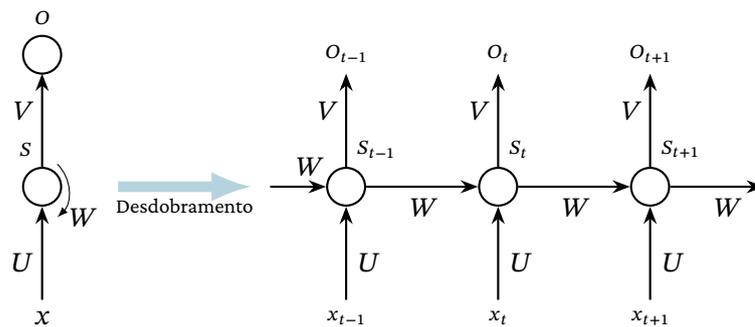
2.3.4.2 *Recurrent Neural Network* – RNN

A principal diferença entre esse tipo de rede e a *Feed Forward* é a realimentação de informações. Para tal, utiliza-se de passos temporais, onde é possível realizar a realimentação de passos anteriores para passos futuros. Dessa forma, redes RNN possuem a habilidade de lidar com dados sequenciais no tempo, além de possuírem uma memória interna possibilitando considerar essas dependências temporais. A disposição dessa rede neural pode ser verificada na Figura 11. Esse tipo de rede possui limitações quando o intervalo sequencial é extremamente longo (LECUN; BENGIO; HINTON, 2015).

2.3.4.2.1 *Long Short Term Memory* – LSTM

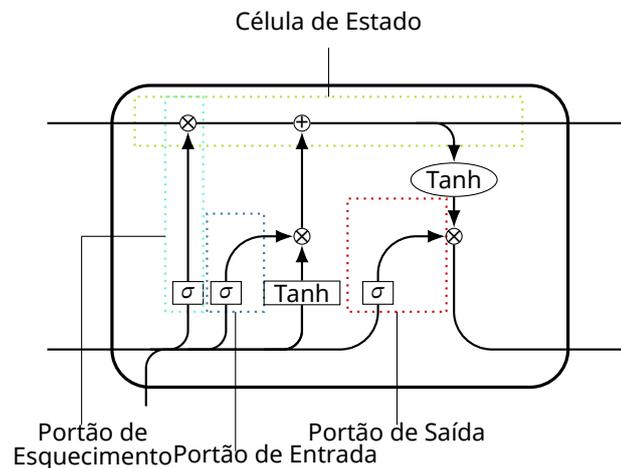
As redes LSTM são um tipo específico da RNN, porém, elas possuem a capacidade de manter as informações na memória interna por mais tempo, sendo uma alternativa para lidar com problemas sequenciais que possuem uma dependência a longo prazo. A estrutura interna dessa topologia pode ser descrita Figura 12.

Figura 11 – Padrão de conexão dos neurônios em uma rede neural de topologia RNN.



Fonte: Adaptado de [Lecun, Bengio e Hinton \(2015\)](#).

Figura 12 – Padrão de conexão dos neurônios em uma rede neural de topologia LSTM.



Fonte: Adaptado de [Mueller e Massaron \(2020\)](#).

Célula de Estado: essa unidade é uma célula de memória, que pode armazenar as informações ao longo do tempo. O fluxo nessa unidade é limitado pelos portões.

Portão de Esquecimento: Essa unidade decide quais informações devem ser descartadas ou mantidas.

Portão de Entrada: controla quais novas informações serão adicionadas à célula de estado. Para tal, utiliza-se das informações da entrada atual e da saída da camada anterior.

Portão de Saída: determina qual será a parte da memória de estado que será enviada como saída de célula LSTM ([OLAH, 2015](#); [HOCHREITER](#); [SCHMIDHUBER, 1997](#)).

3 Física de Partículas

3.1 Modelo Padrão – Uma visão Geral

O Modelo Padrão (MP) é a teoria mais bem comprovada da Física. É baseado na Teoria Quântica de Campos, e descreve três das quatro forças fundamentais: interação forte, interação fraca e eletromagnética. Apesar de esforços contínuos, não foi possível englobar a teoria gravitacional no MP, prevendo como as partículas interagem entre si formando a matéria conhecida atualmente (THOMSON, 2013).

De maneira geral, é possível classificar as partículas fundamentais, partículas construtoras de tudo o que se conhece, em três tipos: quarks, léptons e bósons. As antipartículas são partículas de mesmos números quânticos, mas com cargas elétricas opostas. É possível verificar as partículas elementares na Figura 13, assim como suas antipartículas.

Apesar de atualmente ser uma teoria muito bem fundamentada, por um bom tempo acreditou-se que a quantidade de partículas elementares fosse bem maior. Como esse trabalho está relacionado com a Cromodinâmica Quântica, a teoria que estuda as interações de partículas com cor, ou seja, entre os quarks e sua partícula mediadora, o glúon, o desenvolvimento da descoberta de algumas dessas partículas estará limitado entre elas.

Figura 13 – Partículas elementares do Modelo Padrão.

	Três gerações de matéria (férmions)			Três gerações de antimatéria (antiférmions)			Mediadores (bósons)	
	I	II	III	I	II	III		
massa	$\approx 2.2 \text{ MeV}$	$\approx 1.3 \text{ GeV}$	$\approx 173 \text{ GeV}$	$\approx 2.2 \text{ MeV}$	$\approx 1.3 \text{ GeV}$	$\approx 173 \text{ GeV}$	0	$\approx 125 \text{ GeV}$
carga	$+2/3$	$+2/3$	$+2/3$	$-2/3$	$-2/3$	$-2/3$	0	0
spin	$1/2$	$1/2$	$1/2$	$1/2$	$1/2$	$1/2$	1	0
	u up	c charm	t top	\bar{u} antiup	\bar{c} anticharm	\bar{t} antitop	g glúon	H Higgs
QUARKS	d down	s strange	b bottom	\bar{d} antidown	\bar{s} antistrange	\bar{b} antibottom	γ fóton	Z⁰ bóson Z
	$\approx 4.7 \text{ MeV}$	$\approx 96 \text{ MeV}$	$\approx 4.2 \text{ GeV}$	$\approx 4.7 \text{ MeV}$	$\approx 96 \text{ MeV}$	$\approx 4.2 \text{ GeV}$	0	$\approx 91.2 \text{ GeV}$
	$-1/3$	$-1/3$	$-1/3$	$+1/3$	$+1/3$	$+1/3$	0	1
	$1/2$	$1/2$	$1/2$	$1/2$	$1/2$	$1/2$	1	1
	e⁻ elétron	μ^- múon	τ^- tau	e⁺ elétron	μ^+ múon	τ^+ tau	Z⁰ bóson Z	W⁰ bóson W
LÉPTONS	$< 1.0 \text{ eV}$	$< 0.17 \text{ eV}$	$< 18.2 \text{ MeV}$	$< 1.0 \text{ eV}$	$< 0.17 \text{ eV}$	$< 18.2 \text{ MeV}$	$\approx 80.4 \text{ GeV}$	$\approx 80.4 \text{ GeV}$
	0	0	0	0	0	0	-1	+1
	$1/2$	$1/2$	$1/2$	$1/2$	$1/2$	$1/2$	1	1
	ν_e neutrino do elétron	ν_μ neutrino do múon	ν_τ neutrino do tau	$\bar{\nu}_e$ antineutrino do elétron	$\bar{\nu}_\mu$ antineutrino do múon	$\bar{\nu}_\tau$ antineutrino do tau	W⁻ bóson W	W⁺ bóson W

BÓSONS ESCALARES
BÓSONS DE GAUGE
BÓSONS VETORIAIS

Fonte: Adaptado de Thomson (2013).

3.1.1 Os Quarks

O *Teorema de Noether* foi desenvolvido por Emmy Noether, e teve sua primeira versão finalizada em 1918. De acordo com Einstein, ela foi uma das mais importantes matemáticas que já existiu. Tendo como base na teoria de grupos, esse teorema associa simetrias com leis de conservação físicas (GRIFFITHS, 1987; OLVER, 2018).

Para exemplificação, na Tabela 1 está a relação entre as simetrias e as leis de conservação.

Tabela 1 – Relação entre as simetrias e as leis de conservação.

SIMETRIA	LEI DE CONSERVAÇÃO
Temporal	Energia
Espacial	Momento
Rotacional	Momento Angular
Paridade	Helicidade

Fonte: Adaptado de Oldoni da Silva e Natti (2007) e Olver (2018).

A partir desse teorema, físicos procuram encontrar simetrias com o intuito de determinar leis de conservação. E é assim que se inicia essa seção: procurando uma simetria entre prótons e nêutrons.

Prótons e nêutrons se diferenciam principalmente pela sua carga elétrica e, por isso, foram interpretados como estados diferentes de uma mesma partícula: o núcleon. A diferença de massa entre esses dois estados era justificada pela interação eletromagnética.

Similar à estrutura matemática do spin do elétron, introduziu-se o Isospin como sendo o formalismo matemático da similaridade do próton e nêutron, utilizando as propriedades do grupo SU(2), sendo possível escrevê-los como “múltiplos de carga” de multiplicidade $2t + 1$, sendo $t = 1/2$ no caso de núcleons.. Os geradores desse grupo são as matrizes de Pauli, como mostra a Representação (3.1)

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (3.1)$$

onde o espaço formado por esses geradores é conhecido como *espaço de carga*. Relaciona-se os estados de próton e nêutron aos seus autovetores com autovalores $+1/2$ e $-1/2$, ou seja,

$$\text{próton} \leftrightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \equiv \left| \frac{1}{2}, \frac{1}{2} \right\rangle \quad \text{e} \quad \text{nêutron} \leftrightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \equiv \left| \frac{1}{2}, -\frac{1}{2} \right\rangle. \quad (3.2)$$

Como cada núcleon tem $I = 1/2$ e a terceira projeção do isospin $I_3 = 1/2$, têm-se estados núcleon-núcleon se $I = 1$ e $I = 0$, tais que

$$\begin{cases} |I = 1, I_3 = 1\rangle = \left| \frac{1}{2}, \frac{1}{2} \right\rangle \otimes \left| \frac{1}{2}, \frac{1}{2} \right\rangle = |p\rangle |p\rangle, \\ |I = 1, I_3 = 0\rangle = \frac{1}{\sqrt{2}}(pn + np), \\ |I = 1, I_3 = -1\rangle = \left| \frac{1}{2}, -\frac{1}{2} \right\rangle \otimes \left| \frac{1}{2}, -\frac{1}{2} \right\rangle = |n\rangle |n\rangle, \end{cases} \quad (3.3)$$

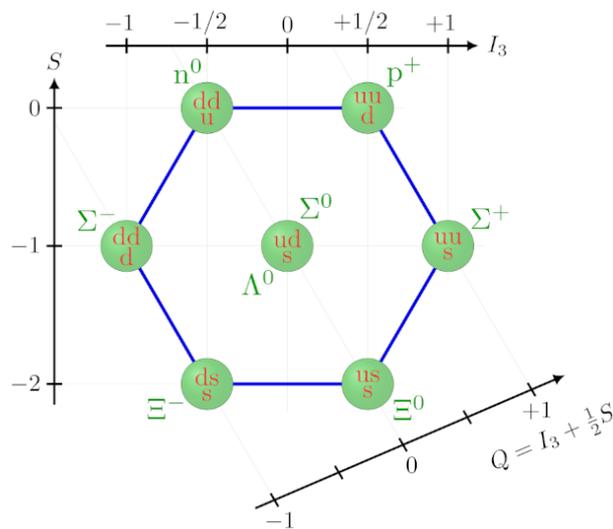
e o singlete

$$|I = 1, I_3 = 0\rangle = \frac{1}{\sqrt{2}}(pn - np). \quad (3.4)$$

Apesar do modelo ter explicado muito bem alguns fenômenos que aconteciam entre prótons e nêutrons, a partir da década de 1950, com a criação de colisores de partículas, a quantidade de partículas descobertas aumentou grandemente. Contudo, algumas apresentavam um comportamento não esperado: elas decaíam em um tempo mais longo do que o calculado (em torno de 1×10^{-10} s), contudo quando eram produzidas seu tempo de vida esperado era curto: 1×10^{-23} s. Gell-Mann ao estudar essas propriedades, teorizou um novo número quântico: a *estranheza*.

Considerando esse novo número quântico, Gell-Mann conseguiu, em 1961, organizar as partículas até então conhecidas no que ficou conhecido como *caminho óctuplo*. É possível verificar a organização proposta por Gell-Mann, e independentemente por Yuval Ne'eman, na Figura 14 (MOREIRA, 2007).

Figura 14 – Caminho óctuplo dos bárions.



Fonte: Adaptado de Griffiths (1987).

Gell-Mann, em 1964, propôs o *modelo a quarks*, utilizando o grupo $SU(3)$, supondo que, na realidade, existiam três constituintes da matéria, que ele denominou quarks: o *up* (u), o *down* (d) e o *strange* (s). Eis aqui uma justificativa pelo sucesso da Teoria do Isospin: na realidade tratava-se da simetria entre os quarks u e d , tais quais possuem massas muito próximas, como pode ser verificado na Figura 13. Logo, têm-se

$$\text{quark up} = |1/2, +1/2\rangle, \quad \text{sendo } I = 1/2 \text{ e } I_3 = +1/2,$$

$$\text{quark down} = |1/2, -1/2\rangle, \quad \text{sendo } I = 1/2 \text{ e } I_3 = -1/2.$$

Portanto, para os tripletos com $I = 1$, têm-se:

$$|1, 1\rangle = |u\rangle |u\rangle,$$

$$|1, 0\rangle = \frac{1}{\sqrt{2}}(ud + du),$$

$$|1, -1\rangle = |d\rangle |d\rangle.$$

Considerou-se que cada quark, até então como o estipulado por Gell-Mann, teria número bariônico e número de estranheza como pode ser verificado na Tabela 2.

Tabela 2 – Número bariônico e estranheza para os quarks up, down, strange e os antiquarks.

NÚMERO QUÂNTICO	u	d	s	\bar{u}	\bar{d}	\bar{s}
Bariônico (B)	+1/3	+1/3	+1/3	-1/3	-1/3	-1/3
Estranheza (S)	0	0	-1	0	0	+1

Fonte: Adaptado de Griffiths (1987).

Definiu-se como hipercarga a soma do número bariônico com a estranheza

$$Y = B + S, \quad (3.5)$$

e ainda relacionou a carga de um carga com a hipercarga, como pode ser visto na Equação (3.6)

$$Q = I_3 + \frac{Y}{2}. \quad (3.6)$$

Agora tendo como base o grupo SU(3), as matrizes geradoras ficaram conhecidas como matrizes de Gell-Mann, sendo elas:

$$\begin{aligned} \gamma_1 &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, & \gamma_2 &= \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, & \gamma_3 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, & \gamma_4 &= \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \\ \gamma_5 &= \begin{pmatrix} 0 & 0 & -i \\ 0 & 0 & 0 \\ i & 0 & 0 \end{pmatrix}, & \gamma_6 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, & \gamma_7 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix}, & \gamma_8 &= \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}. \end{aligned}$$

As representações fundamentais do grupo SU(3)

$$u = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad d = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad s = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Além das cargas fracionárias, outro problema que ocorria com essa representação era a diferença de massa entre os quarks. O quark s possuía uma massa cerca de 90 MeV maior quando comparado com a massa dos quarks u e d , o que fazia com que a simetria SU(3) para o sabor dos quarks fosse uma aproximação, ou seja, não exata (GONÇALVES, J. T. S., 2017).

Com esse novo arranjo, considerando que as partículas possuíam os blocos construtores agora chamados de quarks, Gell-Mann as organizou em uma nova disposição de cargas. Um fato que chamou a atenção, além da demora da comunidade científica em aceitar os quarks, era que eles possuíam cargas fracionárias (GRIFFITHS, 1987).

Apesar dessa peculiar classificação com carga elementar fracionada, o modelo de quarks obteve grande sucesso quando conseguiu prever partículas descobertas posteriormente. No entanto,

surgiu um problema com a descoberta da partícula Ω^- , composta por três quarks *strange*, o que parecia violar o princípio da exclusão de Pauli. Para contornar essa questão, Boris Struminsky propôs a necessidade de um número quântico adicional para os quarks, a fim de evitar a violação do princípio de Pauli.

Um problema semelhante ocorria com a partícula Δ^{++} , composta por três quarks *up*. Em 1965, Moo-Young Han e Yoichiro Nambu propuseram que os quarks possuíssem um grau de liberdade adicional baseado no grupo de Gauge SU(3), que foi chamado de *cor*. A partir dessa propriedade, Han e Nambu perceberam que os quarks interagem por meio de um octeto de bósons de gauge conhecidos como glúons (THOMSON, 2013). Assim, surgiu a Cromodinâmica Quântica (QCD), a qual é a teoria quântica de campos responsável pela interação forte. Atualmente, foram descobertos 6 quarks, os quais podem ser verificados na Figura 13, assim como os antiquarks. Acredita-se que não existem outros quarks além dos já encontrados (OLDONI DA SILVA; NATTI, 2007).

3.1.2 Cromodinâmica Quântica

O desenvolvimento da QCD tem sua origem no sucesso da Eletrodinâmica Quântica (QED). Na QED, o grupo que descreve a simetria do universo é o grupo U(1), enquanto na QCD o grupo associado é o grupo SU(3) (THOMSON, 2013). Essa simetria, SU(3), é estabelecida por meio de transformações locais de fase, conforme descrito pela Equação de Dirac,

$$i\gamma^\mu \partial_\mu \psi = m\psi, \quad (3.7)$$

que continuaria invariante mediante uma transformação SU(3)

$$\psi(x) \rightarrow \psi'(x) = \exp [ig_s \alpha \cdot \hat{T}] \psi(x), \quad (3.8)$$

sendo $\hat{T} = T^\alpha$ os geradores do grupo SU(3), que estão relacionados com as matrizes de Gell-Mann por

$$T^\alpha = \frac{1}{2} \gamma^\alpha.$$

Dessa forma ficou evidente que a equação de uma partícula livre não era adequada para a transformação SU(3) mencionada anteriormente. Isso ocorre porque ela resultava em um termo adicional, indicando que a Equação de Dirac, Equação (3.7), não é invariante sob as transformações do SU(3) (THOMSON, 2013).

Devido aos geradores do grupo SU(3) serem matrizes de ordem 3, torna-se evidente que partículas sujeitas a essas transformações devem possuir três graus de liberdade adicionais. Esses graus de liberdade são conhecidos como “cor”. As cores utilizadas nesse contexto são o vermelho, azul e verde (GONÇALVES, J. T. S., 2017).

Do ponto de vista físico, quando um sistema passa por uma transformação do tipo SU(3), ele está na verdade, percorrendo seu espaço de cor. Portanto, para que a transformação descrita na Equação (3.8) seja respeitada, a Equação de Dirac é modificada, como mostra a Equação (3.9)

$$i\gamma^\mu [\partial_\mu + i g_s (\partial_\mu \alpha) \cdot \hat{T}] \psi = m\psi. \quad (3.9)$$

Reescrevendo (3.9), considerando $\partial_\mu \alpha = G_\mu^a(x)$, onde $a = 1, \dots, 8$ são os geradores do grupo SU(3), que passam a ser interpretados como campos de gauge, obtêm-se como segue em (3.10)

$$i\gamma^\mu [\partial_\mu + ig_s G_m^a u T^a] \psi - m\psi = 0, \quad (3.10)$$

de forma que os geradores são invariantes, perante a transformação

$$G_\mu^k \rightarrow G_\mu^{k'} = G_\mu^k - \partial_\mu \alpha \mu - g_s f_{ijk} \alpha_i G_\mu^j.$$

A QCD é uma teoria não abeliana, o que significa que seus geradores não comutam entre si. Isso implica que o termo $g_s f_{ijk} \alpha_i G_\mu^j$ causa uma interação entre os próprios geradores. Para esclarecer, a ordem dos geradores do SU(3) resulta nas diferentes cores, e esses geradores são responsáveis pelas interações entre os quarks e entre os próprios geradores. Esses geradores são identificados como *glúons*, sendo também as partículas mediadoras da interação forte. Agora, o foco está na interação entre glúons e quarks. O vértice de interação quark-quark-glúon é descrito pela Equação (3.11)

$$g_s T^a \lambda^\mu G_\mu^a \psi = g_s \frac{1}{2} \gamma^\alpha \lambda^\mu G_\mu^a \psi, \quad (3.11)$$

onde g_s representa a constante de acoplamento da interação forte, e γ são as matrizes de Dirac, como mostra a Equação (3.7) (THOMSON, 2013; GONÇALVES, J. T. S., 2017; GRIFFITHS, 1987). Para surpresa geral a interação forte se comportava de maneira contrária às outras interações conhecidas; quanto mais próximos os quarks uns dos outros, menos intensa é a interação e vice-versa. Este fenômeno levou a dois estados: confinamento e não confinamento de quarks e glúons, abordados na próxima seção.

A utilização de potenciais é um método aplicado no estudo da interação forte, permitindo uma visão mais realista do problema. Para a interação entre quarks e glúons, o potencial é separado em duas partes: um regime que trata do confinamento, e outro que trata do não confinamento. O regime de confinamento é definido, utilizando a linguagem de potencial, como

$$V(\vec{r}) \propto \kappa r,$$

indicando que quanto maior a distância entre as partículas, maior será a energia necessária para separá-las. Esse fenômeno é conhecido como *confinamento*. Por outro lado, quando as partículas estão muito próximas, elas se comportam como partículas livres. Essa propriedade é conhecida como *liberdade assintótica*. Em resumo, o confinamento implica que as partículas estão confinadas em regiões pequenas, enquanto a liberdade assintótica ocorre quando as partículas estão próximas o suficiente para se comportarem como partículas livres (THOMSON, 2013; DEUR; BRODSKY; DE TÉRAMOND, 2016; GROSS, 1999).

Com base nesse fato, é possível dividir a QCD em dois regimes: o regime *perturbativo* e o regime *não perturbativo*. O regime perturbativo abrange a faixa de energia em que $Q > 1$ GeV, enquanto o regime não perturbativo abrange a faixa de energia em que $Q < 1$ GeV. No regime não perturbativo, utiliza-se a QCD em rede, a qual é um método computacional que discretiza o espaço-tempo em uma grade (DEUR; BRODSKY; DE TÉRAMOND, 2016).

Assim como a constante de estrutura fina na QED, a constante de acoplamento na QCD também precisa passar pelo processo de renormalização. Isso significa que, para níveis de energia acima da escala de confinamento, ou seja, no regime perturbativo, têm-se

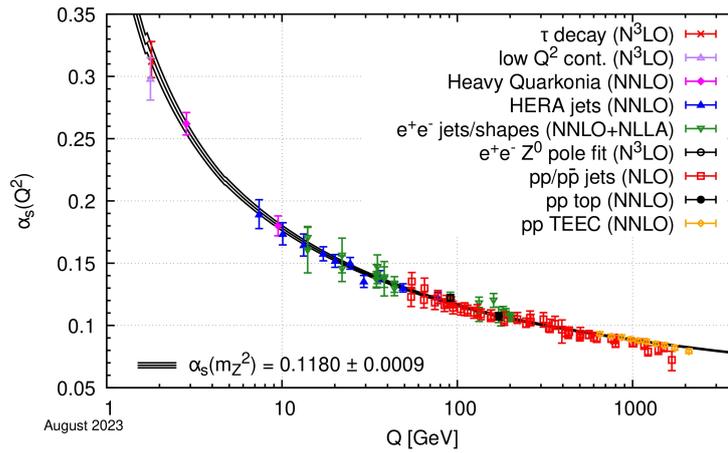
$$\alpha_s(Q^2) = \frac{\alpha_s(\mu^2)}{1 + \beta\alpha_s(\mu^2) \ln\left(\frac{Q^2}{\mu^2}\right)} \quad (3.12)$$

onde

$$\beta = \frac{11N_c - 2N_f}{12\pi}, \quad (3.13)$$

e N_c é o número de cores, N_f é o número de sabores, e $\alpha_s(\mu^2)$ é o valor da constante de acoplamento já conhecida para o nível de energia μ . Para os valores perturbativos, a Figura 15, mostra os valores da constante de acoplamento.

Figura 15 – Constante de acoplamento da QCD.



Fonte: Retirado de [Group et al. \(2020\)](#).

Devido à dificuldade do regime não perturbativo, as teorias que descrevem a constante de acoplamento nesse regime apresentam divergências. Para lidar com essa questão, foram selecionadas duas teorias entre várias opções. A primeira delas é a teoria *Light Front Holographic*, (em tradução literal do Inglês, *luz frontal holográfica*) que pode ser resumida pela expressão apresentada na Equação (3.14)

$$\frac{\alpha_s^{\text{LFH}}(Q^2)}{\pi} = \exp\left(-\frac{Q^2}{4\kappa^2}\right), \quad (3.14)$$

onde $\kappa = M_P/2$, sendo M_P a massa do próton ([DEUR; BRODSKY; TÉRAMOND, 2017](#)).

Nesse caso, pode-se observar que a constante de acoplamento segue um comportamento gaussiano, sendo contínua no polo de Landau. O polo de Landau corresponde ao ponto de transição entre o regime perturbativo e o não perturbativo, que ocorre em $Q = 1$ GeV. Essa continuidade é favorável, uma vez que se trata de uma região de transição fictícia, pois o regime não perturbativo ainda não é completamente compreendido.

A segunda teoria utilizada, baseada na *Light-Front Approach*, (em tradução literal do Inglês, *abordagem da luz frontal*), é a *Brodsky-de Téra mond-Deur (BdTD)*, dada pela expressão

$$\alpha_s^{\text{BdTD}} = \exp\left(-\frac{Q^2}{4\kappa^2}\right), \quad (3.15)$$

onde $\kappa = 0,5 \text{ GeV}$ (CAMPOS, 2021).

Devido à importância de compreender propriedades fundamentais da QCD, como o confinamento de cor, torna-se essencial o estudo da constante de acoplamento. Além disso, a constante de acoplamento é crucial para entender a ocorrência da restauração da simetria quiral no regime não perturbativo. Esses estudos são fundamentais para aprofundar nosso conhecimento sobre a QCD (CAMPOS, 2021).

4 Metodologia

A metodologia divide-se em três partes: geração dos dados para treinamento e validação, implementação computacional e análise de resultados.

4.1 Geração dos dados

Os dados foram criados a partir das Equações (3.12) a (3.15), respeitando os níveis de energia nos regimes perturbativos e não perturbativos. No regime perturbativo, considerou-se o intervalo do momento transferido sendo $1 \text{ GeV} \leq Q_i \leq 15 \text{ TeV}$, os quais foram discretizados em n níveis, sendo que para cada um havia um valor da constante de acoplamento correspondente, alimentando o próximo nível (FUJIMOTO; FUKUSHIMA; MURASE, 2020). Já para o regime não perturbativo, o intervalo considerado foi $1 \text{ MeV} \leq Q_i \leq 990 \text{ MeV}$. Para ambos os casos $n = 10^3$. A Equação (4.1) define a entrada perturbativa

$$\alpha_s^i(Q_i^2) = \frac{4\pi}{\beta_0 \ln\left(\frac{Q_i^2}{\Lambda^2}\right)}. \quad (4.1)$$

A Equação (4.2) corresponde à Equação (3.14), e a Equação (4.3), corresponde à Equação (3.15), correspondentes aos regimes não perturbativos

$$\frac{\alpha_s^{\text{LFH}}(Q_i^2)}{\pi} = \exp\left(-\frac{Q_i^2}{4\kappa^2}\right), \quad (4.2)$$

$$\alpha_s^{\text{BdTD}} = \exp\left(-\frac{Q_i^2}{4\kappa^2}\right). \quad (4.3)$$

Os dados gerados a partir da Equação (4.1) foram utilizados como dados de treinamento da rede, e os dados gerados a partir das Equações (4.2) e (4.3) compuseram o conjunto de dados de teste.

4.2 Implementação computacional

A implementação foi feita na linguagem de programação Python¹, com a utilização as seguintes bibliotecas:

Numpy para a manipulação de elementos algébricos como vetores e matizes (HARRIS et al., 2020).

Matplotlib para a visualização de resultados (HUNTER, 2007).

¹ <https://www.python.org>

Pandas para a manipulação dos dados gerados (MCKINNEY, 2010; TEAM, 2020).

Keras para a implementação de algoritmos de redes neurais (CHOLLET, 2017).

Scikit-Learn para a implementação de algoritmos de redes neurais e de funcionalidades gerais de aprendizado de máquinas (COURNAPEAU, 2007).

TensorFlow para a implementação de algoritmos de redes neurais e de funcionalidades gerais de aprendizado de máquinas (TENSORFLOW..., 2015).

Para a implementação, optou-se por utilizar o [Google Colab](#), sendo utilizadas as versões das bibliotecas de acordo com a Tabela 3.

Tabela 3 – Versões utilizadas de acordo com cada biblioteca utilizada na implementação das redes neurais.

BIBLIOTECA	VERSÃO
Python	3.10.12
Numpy	1.23.5
Pandas	1.5.3
Keras	2.14.0
Scikit-Learn	1.2.2
TensorFlow	2.14.0

Fonte: A autora.

As redes do tipo *Feed Forward* foram testadas com as topologias presentes na Tabela 4, enquanto para as redes do tipo LSTM, utilizaram-se as topologias presentes na Tabela 5.

Tabela 4 – Descrição dos parâmetros usados para formar as topologias testadas neste trabalho. Foram combinadas três funções de ativação, três quantidades de camadas e quatro quantidades de neurônios por camada.

FUNÇÃO DE ATIVAÇÃO	CAMADAS	NEURÔNIOS
ReLU	5	10
Sigmoide	10	20
Tanh	20	50
	30	100

Fonte: A autora.

Foram testadas todas as possíveis combinações entre funções de ativação, camadas e neurônios, no caso da *Feed Forward*, e todas as possíveis combinações entre funções de ativação, janelas e camadas, no caso da LSTM.

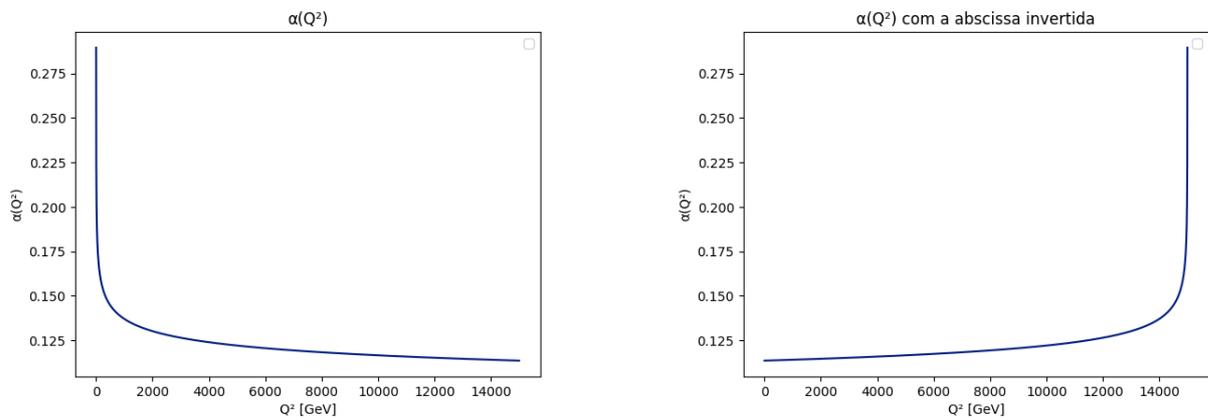
Para as implementações da rede do tipo *Feed Forward*, com a função de ativação do tipo ELU, realizou-se a inversão da abscissa, como mostra a Figura 16. Ela também foi utilizada para a implementação da rede do tipo LSTM, para todas as topologias testadas.

Tabela 5 – Topologias utilizadas para uma rede do tipo LSTM, onde os dados de treinamento foram uma junção das Equações (4.1) e (4.2), para 50 neurônios.

FUNÇÃO DE ATIVAÇÃO	JANELAS	CAMADAS
ReLU	3	3
ELU	5	2
	7	

Fonte: A autora.

Figura 16 – Constante de acoplamento $\alpha(Q^2)$ para a região perturbativa com a abscissa normal em (a) e abscissa invertida em (b).



(a) Abscissa original.

(b) Abscissa invertida.

Fonte: A autora.

Uma breve amostra de como criar uma rede neural em Python pode ser visto na Listagem 1. Nela é criada uma rede com uma camada de entrada e uma de saída, ambas com um neurônio cada, além de cinco camadas ocultas com dez neurônios cada. A função de ativação é a ReLU (2.3.3.1).

4.3 Análise de Dados

Foram realizadas extrapolações para a região não perturbativa, então a análise de dados será considerando os valores da constante de acoplamento da região não perturbativa, Equações (3.14) e (3.15). A partir das previsões os dados serão comparados considerando os valores esperados pela Equação (4.2) e (4.3), considerando as seguintes métricas de erro, onde n é a dimensão do conjunto de dados.

- Erro Quadrático Médio (MSE)

$$\text{MSE} = \frac{1}{n} \sum_{i=0}^n (y_i - \hat{y}_i)^2.$$

- Erro Absoluto Médio (MAE)

$$\text{MAE} = \frac{1}{n} \sum_{i=0}^n \|y_i - \hat{y}_i\|.$$

- Erro Percentual Absoluto Médio (MAPE)

$$\text{MAPE} = \frac{1}{n} \sum_{i=0}^n \left\| \frac{y_i - \hat{y}_i}{y_i} \right\| \times 100\%.$$

Em todos os casos y_i representa o valor esperado e \hat{y}_i representa o valor previsto da constante de acoplamento.

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.python.keras import (Sequential, Model)
from tensorflow.python.keras.layers import Dense
# Modelo de rede neural utilizada
# ativacao = função de ativação escolhida
# camadas = número de camadas ocultas
# neuronios = número de neurônios por camada oculta
ativacao = 'relu'
camadas = 5
neuronios = 10

# Declaração do modelo
model = Sequential()

# Camada de entrada com um neurônio
model.add(Dense(1, activation=ativacao, input_shape=(1, )))

# Adicionando as camadas ocultas
for i in range(camadas):
    model.add(Dense(neuronios, activation=ativacao))

# Camada de saída com um neurônio
model.add(Dense(1))

# Prepara o modelo para execução com otimizador Adam e função de perda MSE
model.compile(optimizer='adam', loss='mse')
```

Fonte: A autora.

Listing 1 – Exemplo de implementação de um modelo de rede neural com uma camada de entrada, diversas camadas ocultas e camada de saída. A função de ativação escolhida é a ReLU 2.3.3.1.

5 Resultados e Discussão

Como este ainda é um campo de pesquisa em sua infância, não existem parâmetros fortes de comparação. Assim, os resultados aqui obtidos foram comparados com os dados esperados do regime não perturbativo, logo, a discussão se restringe ao desempenho da rede considerando o já explicitado.

Os resultados foram divididos em duas principais seções: *Feed Forward* (Seção 5.1) e LSTM (Seção 5.2), onde em cada uma delas utilizou-se um tipo de redes neurais.

5.1 *Feed Forward*

Inicialmente, os dados foram treinados considerando diferentes topologias, variando o número de camadas e a quantidade de neurônios, para três funções de ativação: ReLU, Sigmoide e Tanh. As escolhas para os parâmetros combinados podem ser verificados na Tabela 6.

Tabela 6 – Descrição dos parâmetros usados para formar as topologias testadas neste trabalho. Foram combinadas três funções de ativação, três quantidades de camadas e quatro quantidades de neurônios por camada.

FUNÇÃO DE ATIVAÇÃO	CAMADAS	NEURÔNIOS
ReLU	5	10
Sigmoide	10	20
Tanh	20	50
	30	100

Fonte: A autora.

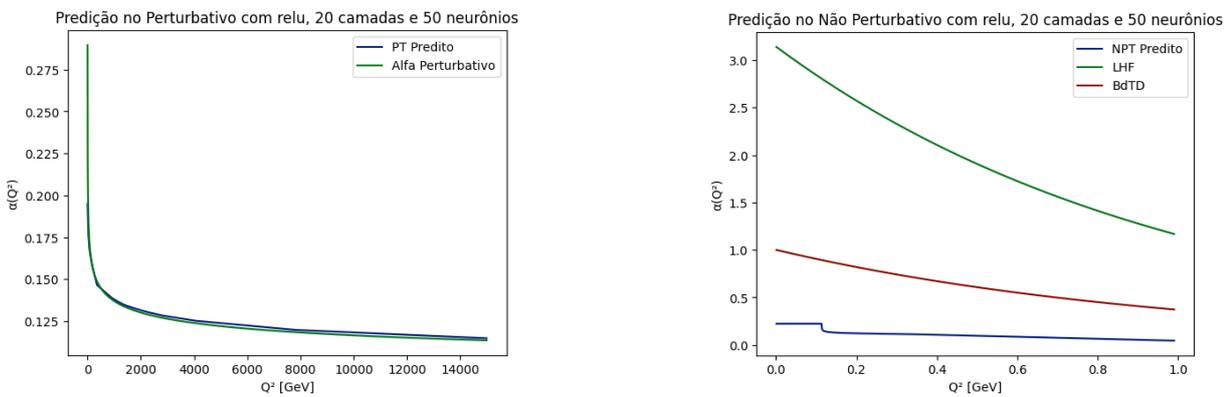
Foram utilizadas 150 épocas, o otimizador ADAM (baseado no gradiente descendente) e a função de perda erro quadrático médio (MSE). Para avaliação da melhor topologia, considerou-se o erro percentual absoluto médio (MAPE), analisado para o regime perturbativo, Equação (4.1), e para o regime não perturbativo, Equações (4.2) e (4.3). A Tabela 7 mostra as melhores topologias considerando a Equação (4.3). É possível verificar graficamente as extrapolações realizadas nas Figuras 17, 18 e 19.

Tabela 7 – Melhores topologias encontradas a partir da métrica MAPE, considerando a Equação (4.3).

FUNÇÃO DE ATIVAÇÃO	CAMADAS	NEURÔNIOS	MAPE – BdTD	MAPE – PERTURBATIVO
Relu	20	50	0,724919	0,843013
Sigmoide	5	50	0,772617	0,617611
Tanh	10	50	0,769980	0,619814

Fonte: A autora.

Figura 17 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta pela função de ativação ReLU, 20 camadas e 50 neurônios.

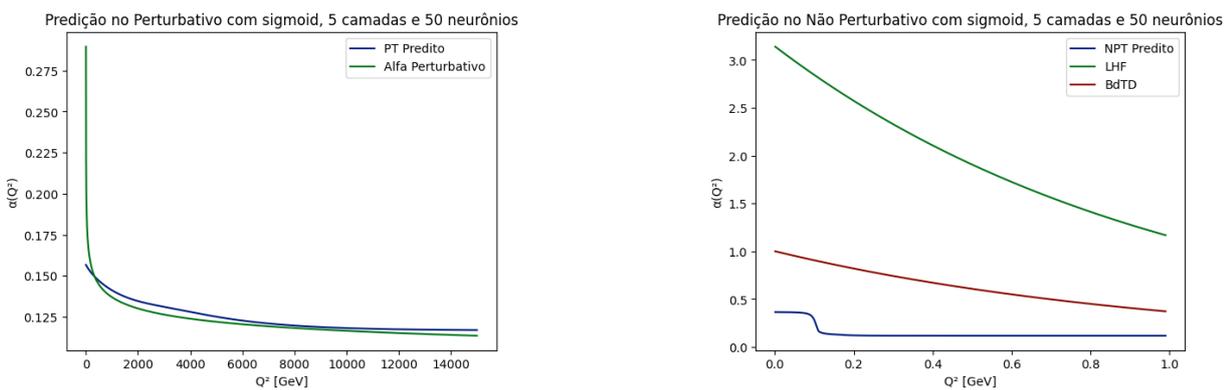


(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.

(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Figura 18 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação sigmoide, 5 camadas e 50 neurônios.



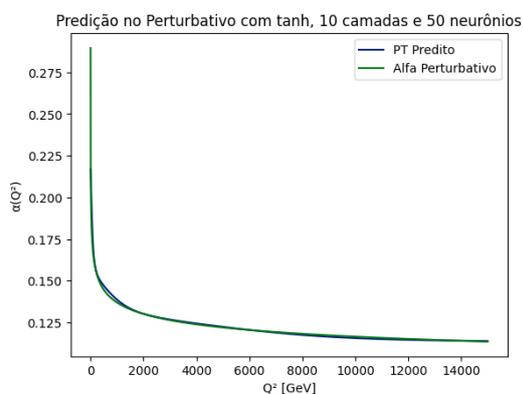
(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.

(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

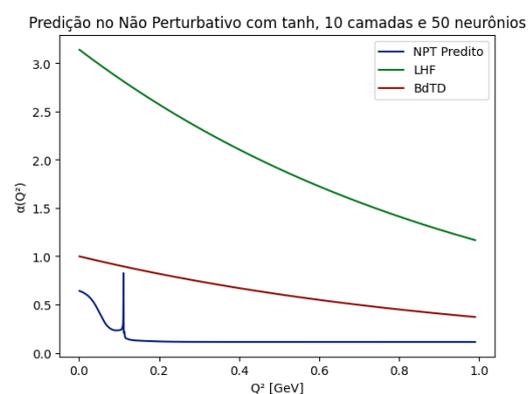
Fonte: A autora.

O desempenho da rede no regime perturbativo foi promissor, sendo minimamente satisfatório no regime não perturbativo. Deve-se observar que nesse caso a rede está tentando prever o comportamento no regime não perturbativo a partir de pouquíssimas informações iniciais e sem

Figura 19 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação tangente hiperbólica, 10 camadas e 50 neurônios.



(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.



(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

a introdução de elementos específicos da física do problema. Algum resultado começa a aparecer próximo à 0 GeV, mas ainda distante das duas teorias de comparação.

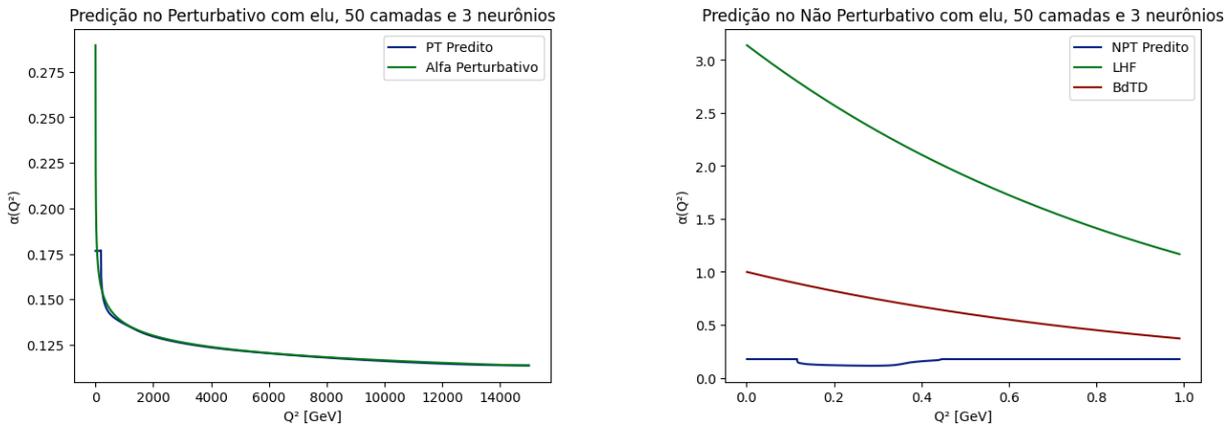
As simulações anteriores levaram à decisão de realizar o teste com outra função de ativação, optando por uma que com características parecidas com a função utilizada no treinamento, levando à escolha da função ELU. As topologias testadas estão na Tabela 8, com 150 épocas, otimizador ADAM e função de perda MSE. As melhores topologias podem ser encontradas na Tabela 8, com gráficos nas Figuras 20 até 25 (métrica MAPE).

Tabela 8 – Melhores topologias encontradas a partir da Métrica MAPE para a função de ativação ELU. As equações utilizadas em cada caso constam da terceira coluna.

CAMADAS	NEURÔNIOS	MAPE – BdTD – EQUAÇÃO (4.3)
50	3	0,718 634
5	3	0,727 368
10	20	0,760 761
CAMADAS	NEURÔNIOS	MAPE – LFH – EQUAÇÃO (4.2)
5	25	0,863 574
3	25	0,869 146
3	15	0,877 070
CAMADAS	NEURÔNIOS	MAPE – PERTURBATIVO – EQUAÇÃO (4.1)
50	3	0,530 809
10	20	0,604 337
5	3	0,624 753

Fonte: A autora.

Figura 20 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação ELU, com 50 camadas e 3 neurônios.

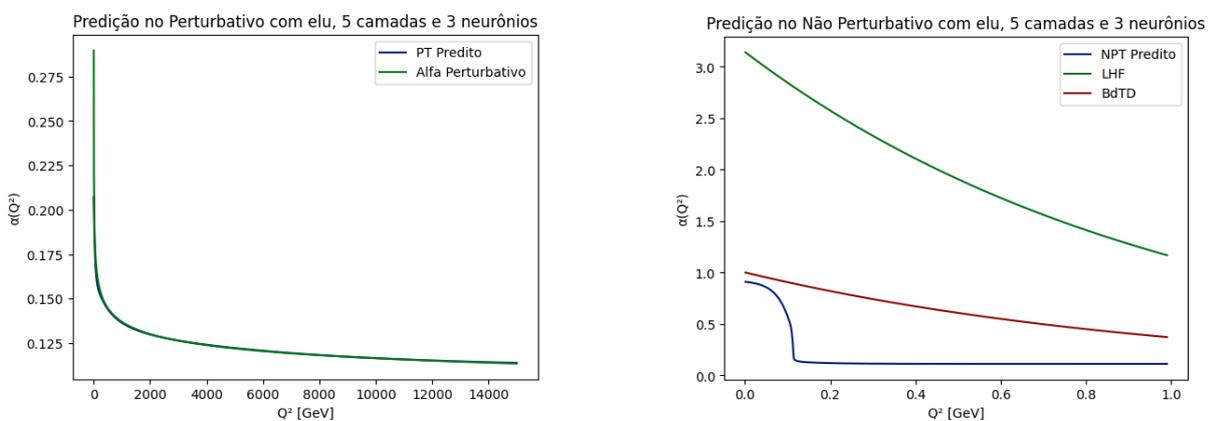


(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.

(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Figura 21 – Extrapolação de $\alpha(Q^2)$ para a região não perturbativa em (a) e perturbativa em (b) com a topologia composta por função de ativação ELU, 5 camadas e 3 neurônios.

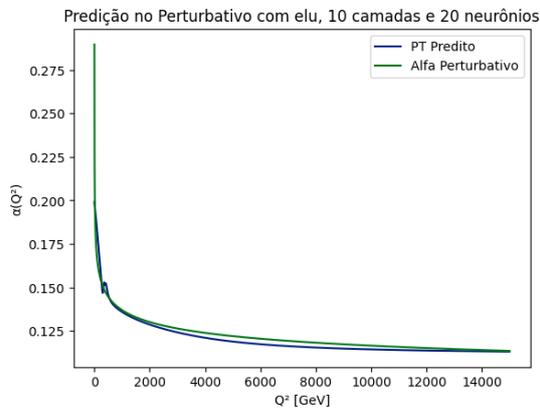


(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.

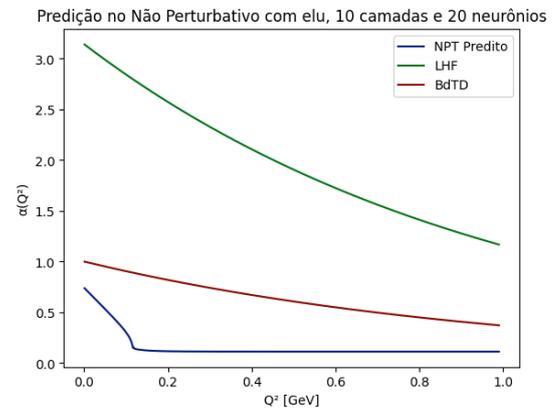
(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Figura 22 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação ELU, 10 camadas e 20 neurônios.



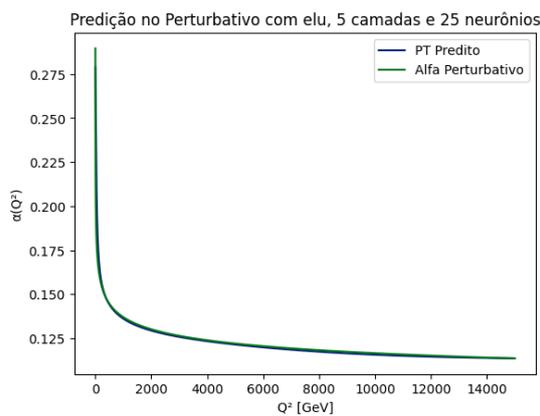
(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.



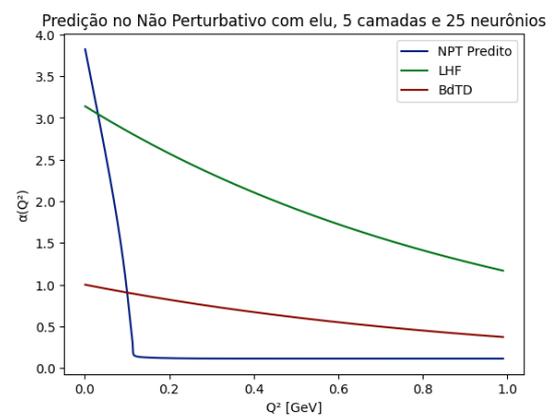
(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Figura 23 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação ELU, 5 camadas e 25 neurônios.



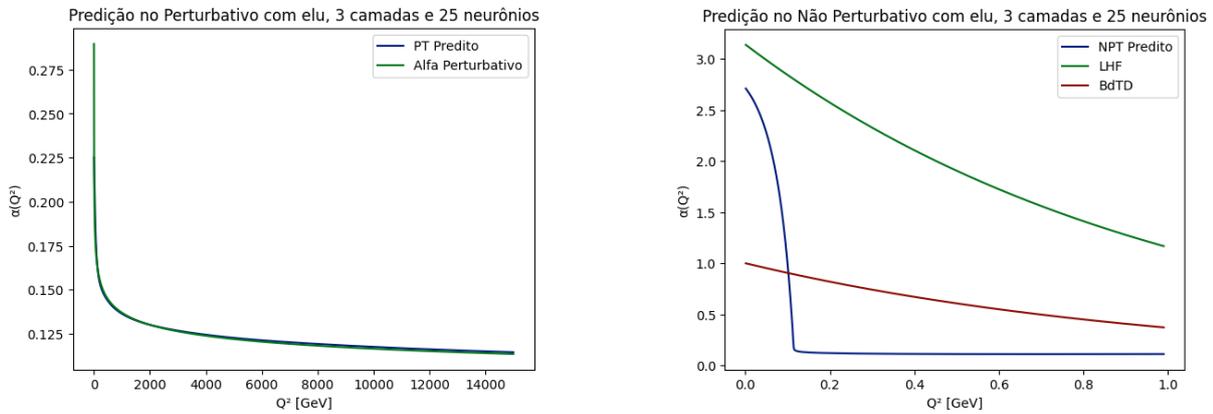
(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.



(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Figura 24 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação ELU, 3 camadas e 25 neurônios.

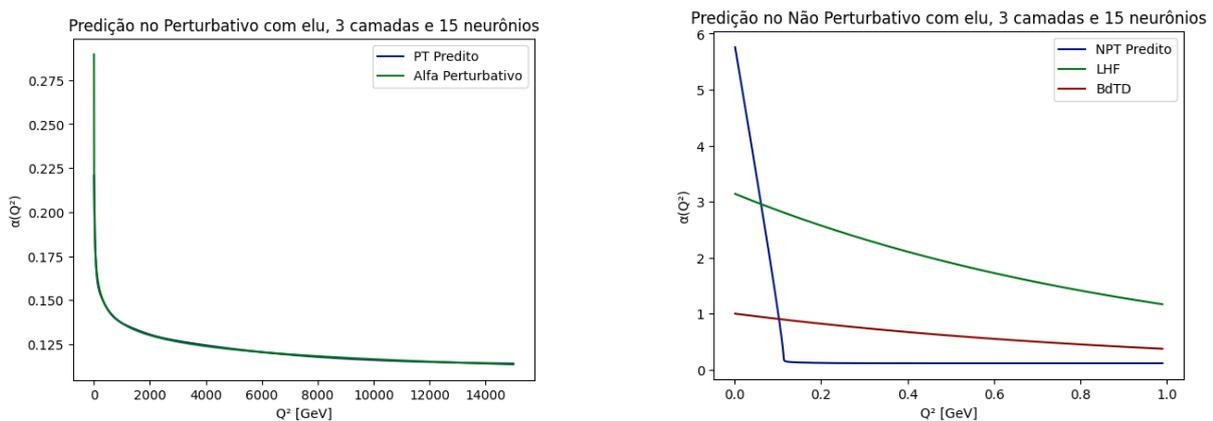


(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.

(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Figura 25 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b) com a topologia composta por função de ativação ELU, 3 camadas e 15 neurônios.



(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.

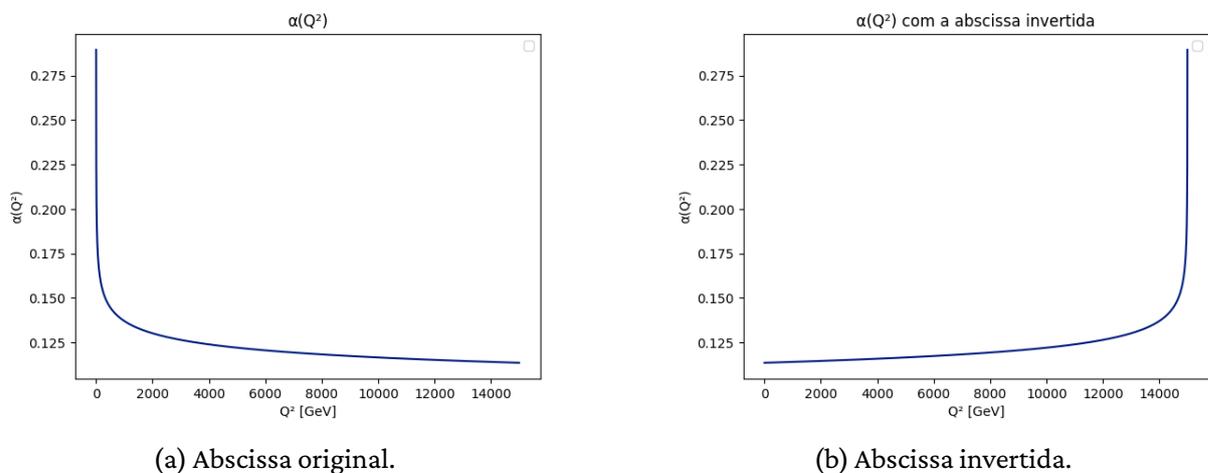
(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Com a função de ativação ELU a rede também começou a apresentar algum tipo de resultado próximo ao 0 GeV de momento trocado. Talvez por sua maior similaridade com as funções que deseja prever, com a ELU conseguiu-se valores maiores de $\alpha(Q^2)$, e portanto mais próximos das duas teorias usadas como comparação.

Como uma possibilidade de tentar melhorar o desempenho da rede, optou-se por inverter a abscissa, como mostra a Figura 26. Tal tática foi utilizada com o intuito de deixar a função mais parecida com a função de ativação.

Figura 26 – Constante de acoplamento $\alpha(Q^2)$ para a região perturbativa com a abscissa normal em (a) e abscissa invertida em (b).



(a) Abscissa original.

(b) Abscissa invertida.

Fonte: A autora.

Os valores da métrica MAPE considerando cada uma das condições, podem ser verificados na Tabela 9, enquanto as Figuras de 27 até 34 mostram as extrapolações dos dados, para o regime perturbativo e para o regime não perturbativo.

Tabela 9 – Melhores topologias encontradas a partir da Métrica MAPE para a função de ativação ELU. As equações utilizadas em cada caso constam da terceira coluna.

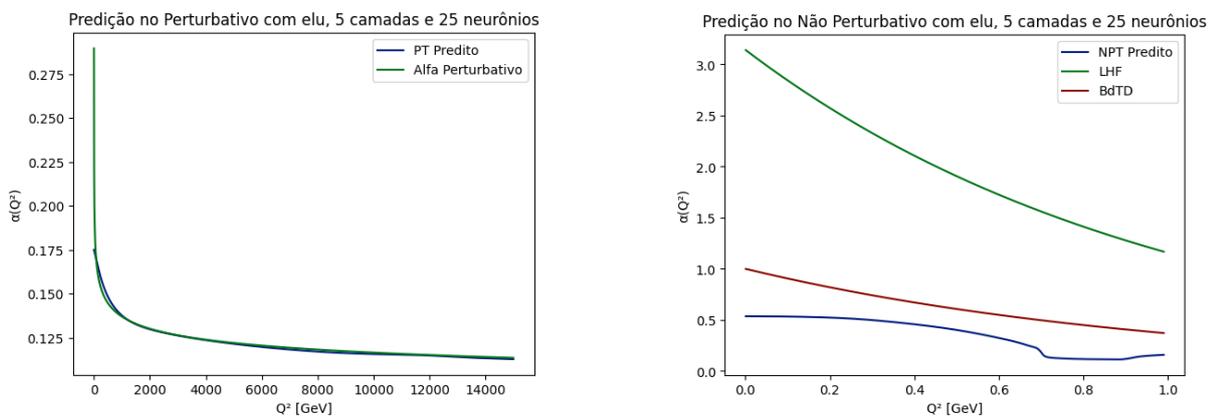
CAMADAS	NEURÔNIOS	MAPE – BbTD – EQUAÇÃO (4.3)
3	3	0,302 886
5	3	0,327 597
5	25	0,474 220

CAMADAS	NEURÔNIOS	MAPE – LFH – EQUAÇÃO (4.2)
3	25	0,369 487
3	20	0,371 200
5	15	0,447 392

CAMADAS	NEURÔNIOS	MAPE – PERTURBATIVO – EQUAÇÃO (4.1)
5	25	0,345 849
50	25	0,473 433
20	15	0,498 860

Fonte: A autora.

Figura 27 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 5 camadas e 25 neurônios.

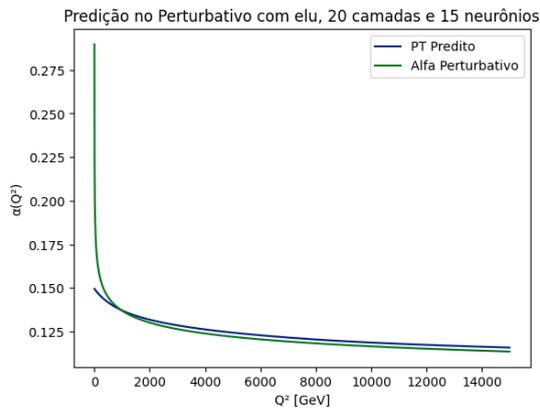


(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.

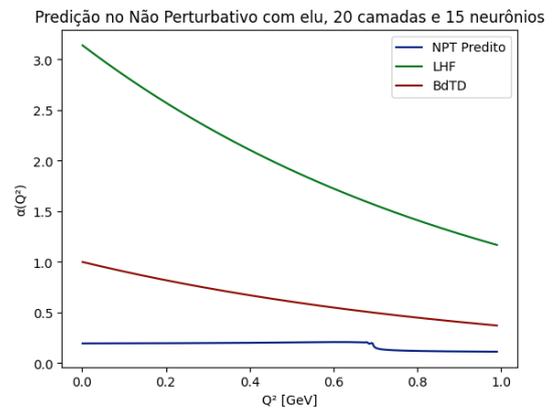
(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Figura 28 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 20 camadas e 15 neurônios.



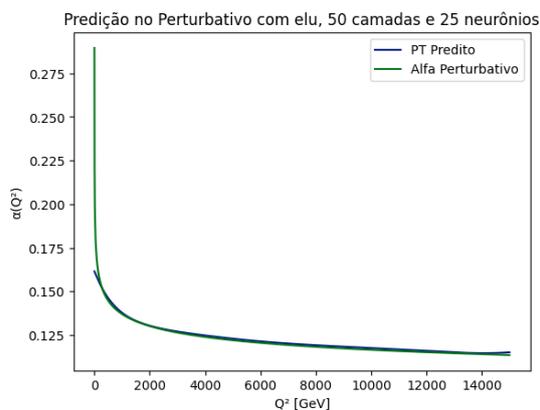
(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.



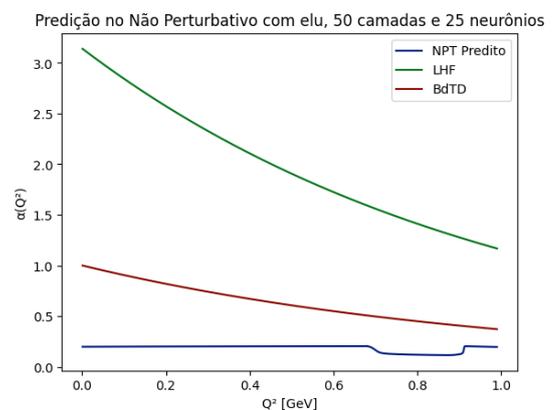
(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Figura 29 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 50 camadas e 25 neurônios.



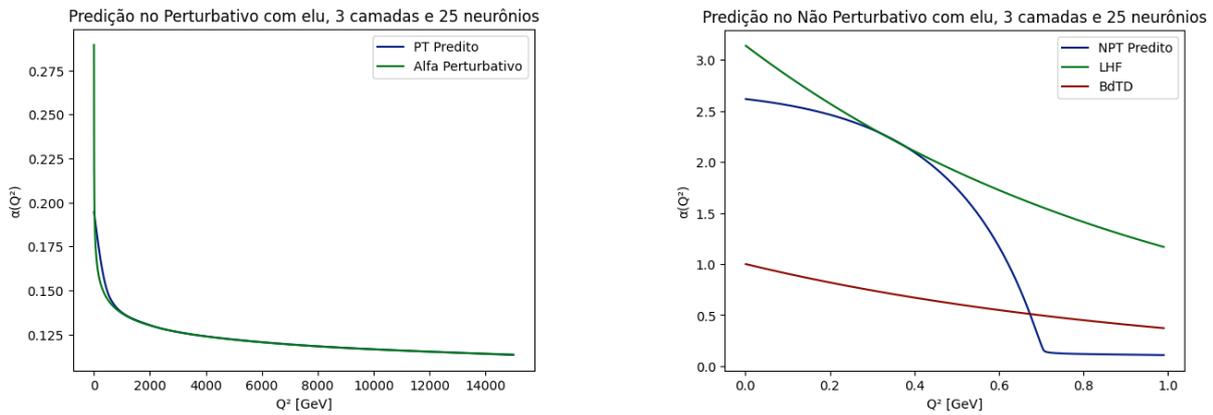
(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.



(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Figura 30 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 3 camadas e 25 neurônios.

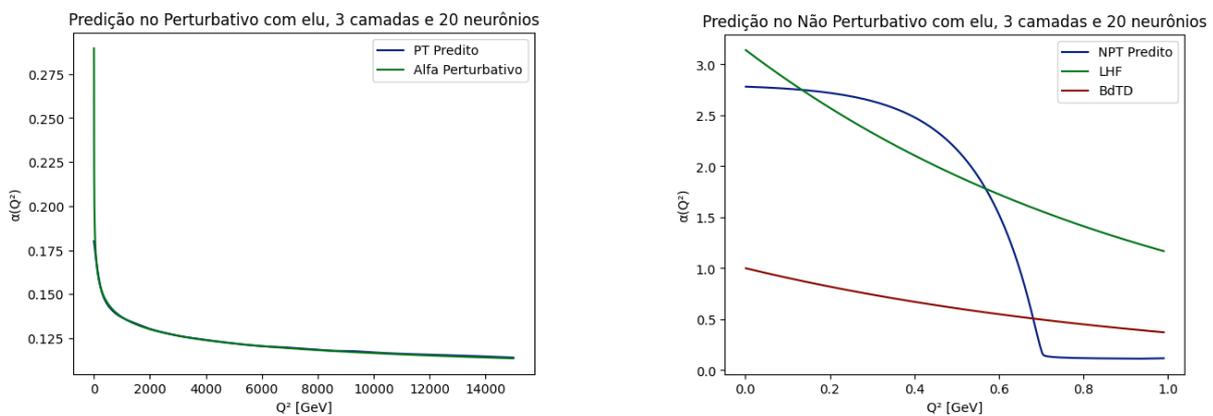


(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.

(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Figura 31 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 3 camadas e 20 neurônios.

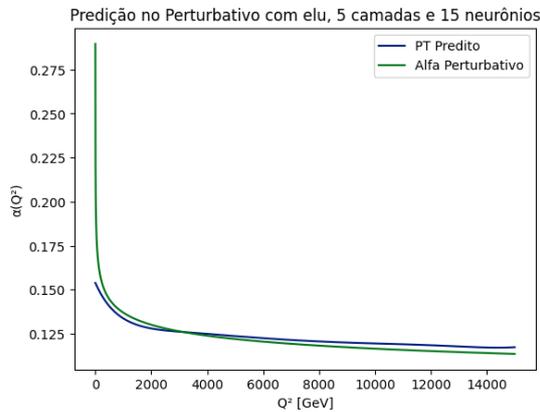


(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.

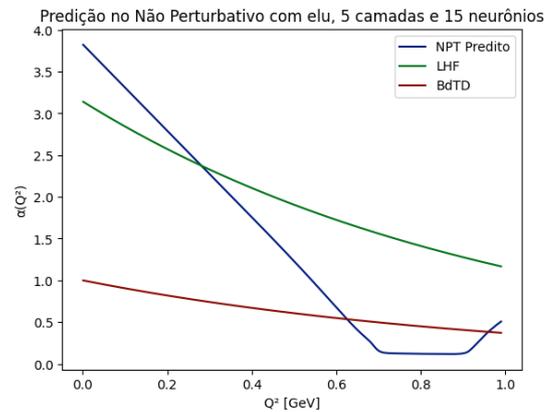
(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Figura 32 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 5 camadas e 15 neurônios.



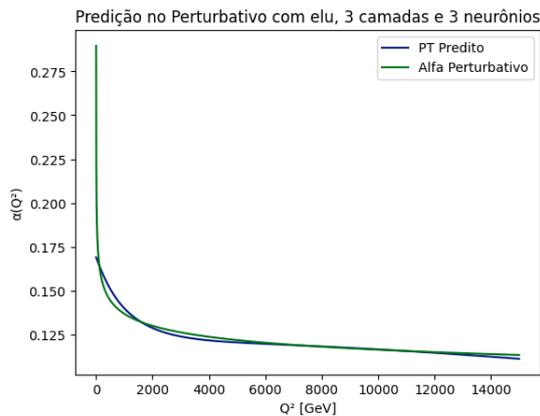
(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.



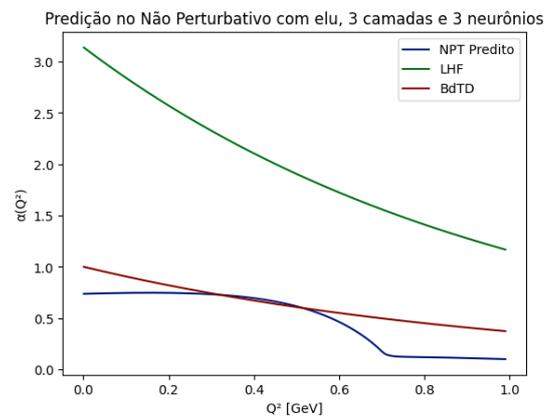
(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Figura 33 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 3 camadas e 3 neurônios.



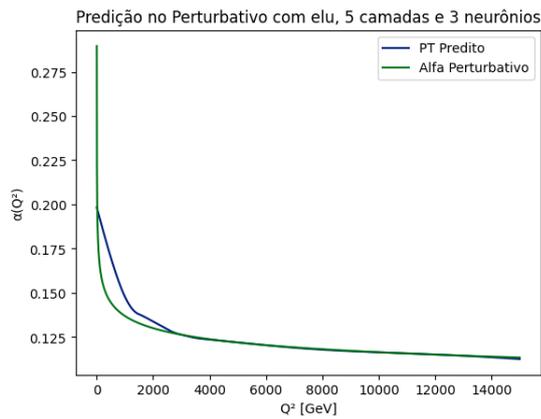
(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.



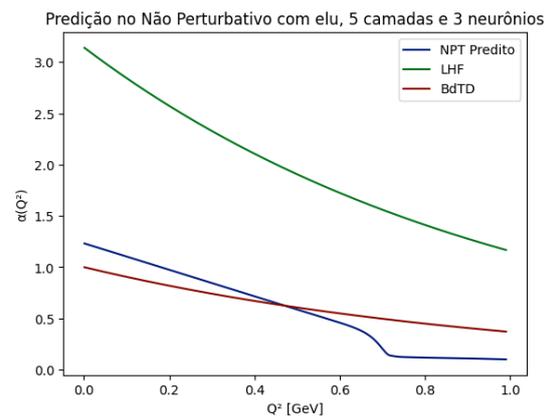
(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

Figura 34 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa em (a) e não perturbativa em (b), com os dados treinados com a abscissa invertida com a topologia composta por função de ativação ELU, 5 camadas e 3 neurônios.



(a) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.



(b) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

Fonte: A autora.

A Tabela 9 mostram uma melhora nas métricas, quando comparadas com os outros métodos utilizados. Para a síntese dos resultados, foi elaborada a Tabela 10, o que permitirá uma discussão um pouco mais clara.

Tabela 10 – Comparativo entre as melhores métricas obtidas para o treinamento normal e para o treinamento com a abscissa invertida, considerando as Equações (4.1), (4.2) e (4.3), para a função de ativação ELU.

	ABSCISSA ORIGINAL	ABSCISSA INVERTIDA
Perturbativa	0,530809	0,345849
LFH	0,863574	0,369487
BdTD	0,718634	0,302886

Fonte: A autora.

Para todas as equações utilizadas como comparação houve uma melhora nos resultados. Pode-se aferir que isso se deu ao fato de que o simples fato de inverter a abscissa, a deixando ainda mais parecida com a função de ativação utilizada, conseguiu melhorar significativamente os valores de métricas obtidos.

Considerando a Figura 27b, verifica-se que o valor extrapolado não acompanha o salto que ocorre perto do 1 GeV. Uma forma de analisar isso é acompanhar a evolução da métrica de erro MAPE, conforme limita-se a quantidade de pontos que são analisados, para o regime perturbativo. O intervalo, assim como os valores da métrica, podem ser verificados na Tabela 11.

Tabela 11 – Evolução da métrica MAPE, para diversos intervalos, considerando a extrapolação no regime perturbativo e a Equação (4.1).

INTERVALO ($Q^2[GeV]$)	MAPE PERTURBATIVO
[1000, 15000]	0,335776
[2000, 15000]	0,315607
[4000, 15000]	0,262906
[8000, 15000]	0,302805
[12000, 15000]	0,228315

Fonte: A autora.

A Tabela 11 mostra como a rede possuiu um desempenho melhor, conforme mais se afastava do regime não perturbativa, mostrando ser promissora para o estudo no regime perturbativo.

Fica claro que a utilização da função de ativação ELU, assim como a inversão da abscissa, possibilitou melhores resultados. A maior dificuldade aparece à medida que se aproxima no regime não perturbativo, devido ao polo de Landau. Para esse caso, como as teorias não são muito bem definidas, opta-se por duas: a LFH e a BdTD. As melhores topologias para essas redes foram, respectivamente, 3 camadas e 25 neurônios, e 3 camadas e 3 neurônios.

5.2 LSTM

Outra possibilidade que surgiu durante a execução deste trabalho é a de considerar os dados como se fossem uma série temporal. Para tal, foi necessário inverter a abscissa, como mostra a Figura 26b. Para séries temporais o tipo de rede mais indicado é a LSTM (HOCHREITER; SCHMIDHUBER, 1997).

Como já foi utilizado a função de ativação ELU e verificou-se que ela foi a que teve melhor desempenho nas redes do tipo *Feed Forward*, optou-se por continuar com essa função de ativação, além de também realizar testes com a função ReLU.

Para a utilização de redes LSTM, é necessário que os dados sejam divididos de uma forma um pouco diferente, denominados *janelas*. As janelas são aplicadas nos conjuntos de dados utilizados para o treinamento e para o teste, derivados da discretização uniforme de $\alpha(Q^2)$. Seja uma série temporal (x_1, x_2, \dots, x_N) , reorganizada em janelas de tamanho J , como mostra a Tabela 12.

Tabela 12 – Relação entre a entrada e a saída com janelas de tamanho J em uma série temporal.

k	x_k	y_k
1	(x_1, \dots, x_J)	x_{J+1}
2	(x_2, \dots, x_{J+1})	x_{J+2}
	\vdots	
$N - J$	$(x_{N-J}, \dots, x_{N-1})$	x_N

Fonte: A autora.

Para o treinamento da rede, combinou-se os dados do domínio perturbativo (Equação (4.1)), com 10% dos dados não perturbativos das Equação (4.2), tentando identificar o “salto” que há na região do polo de Landau. As topologias utilizadas podem ser verificadas na Tabela 13.

Tabela 13 – Topologias utilizadas para uma rede do tipo LSTM, onde os dados de treinamento foram uma junção das Equações (4.1) e (4.2), para 50 neurônios.

FUNÇÃO DE ATIVAÇÃO	JANELAS	CAMADAS
ReLU	3	3
ELU	5	2
	7	

Fonte: A autora.

As melhores topologias, considerando a Equação (4.3) podem ser verificadas na Tabela 14. Os gráficos das melhores topologias estão nas Figuras 35 e 36.

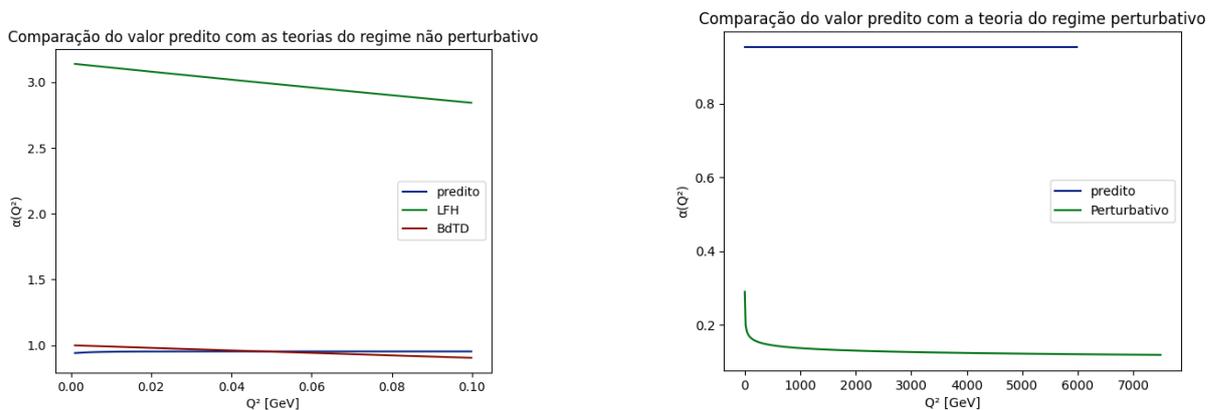
É possível ver que ambas tiveram um resultado satisfatório para o regime não perturbativo, quando considerou-se a Equação (4.3). Contudo, a utilização de dados do regime não perturbativo para o treinamento interferiu o seu desempenho no regime perturbativo. Ainda assim, houve a

Tabela 14 – Melhores topologias encontradas a partir da Métrica MAPE, considerando a Equação (4.3), para 50 neurônios

JANELAS	FUNÇÃO DE ATIVAÇÃO	CAMADAS	MAPE – BdTD	MAPE – PERTURBATIVO
7	ReLU	2	0,025 557	6,375 473
5	ELU	2	0,025 839	6,270 750

Fonte: A autora.

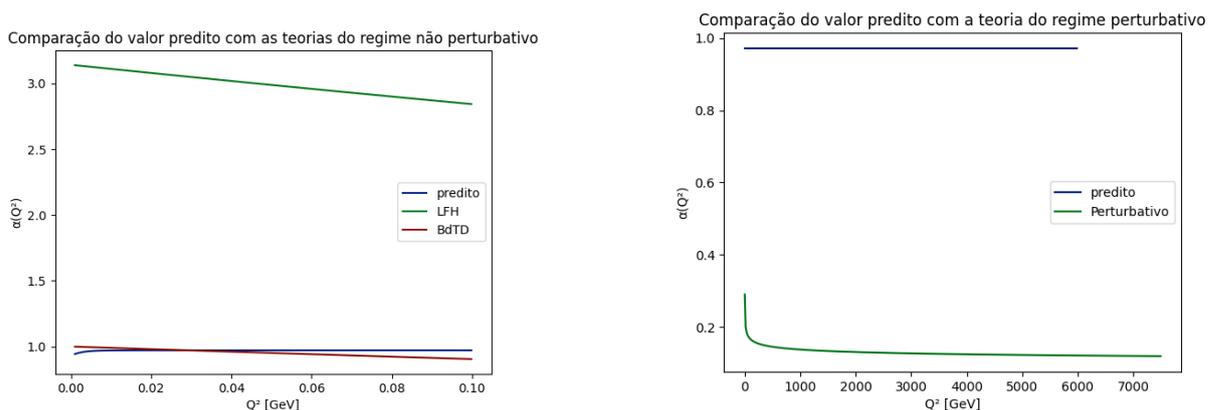
Figura 35 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa e não perturbativa com a topologia composta por função de ativação ReLU, 7 janelas, 2 camadas e 50 neurônios.



(a) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

(b) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.

Figura 36 – Extrapolação de $\alpha(Q^2)$ para a região perturbativa e não perturbativa com a topologia composta por função de ativação ELU, 5 janelas, 2 camadas e 50 neurônios.



(a) Extrapolação de $\alpha(Q^2)$ para a região não perturbativa.

(b) Extrapolação de $\alpha(Q^2)$ para a região perturbativa.

descontinuidade no polo de Landau, mas o que não interfere muito, quando considera-se apenas a comparação entre os modelos.

É possível perceber um padrão: quanto menor o número de camadas, melhor a rede se sai. Isso ocorre também com a análise realizada na rede do tipo *Feed Forward*, que foi possível perceber que, no geral, as redes que não eram tão profundas tiveram um desempenho melhor, em comparação

com as redes mais profundas. Isso pode se dar ao fato de que apesar da não linearidade da função a ser estudada, a aplicação sucessiva de termos não lineares pode se afastar ainda mais da função modelada.

Ao contrário do que foi observado com as redes do tipo *Feed Forward*, as redes do tipo LSTM tiveram um desempenho igualável, quando comparadas as funções de ativação ELU e ReLU. Isso provavelmente se deve ao fato da análise do problema como uma série temporal e a dependência entre os dados.

É necessário reforçar que como os valores dos pesos são inicializados de forma aleatória, os valores obtidos das métricas de avaliação podem variar ao tentar reproduzir. Contudo, o que foi possível observar ao longo da realização deste trabalho é que as redes que se saíram melhores eram as mesmas.

Logo, ambos os tipos de rede se mostraram promissoras para a resolução do problema, contudo, as redes do tipo LSTM, foram mais precisas, quando analisadas a partir da métrica MAPE. Como a QCD não perturbativa ainda é uma teoria incompleta, as possibilidades de estudo da constante de acoplamento, pode ser de grande ajuda para identificar o *confinamento de cor*, considerado um dos problemas do milênio.

6 Conclusões

O presente trabalho possui um caráter exploratório, já que não foram encontrados trabalhos similares na bibliografia. Portanto, os parâmetros de comparação aqui apresentados foram as próprias equações utilizadas para a compreensão do regime não perturbativo da QCD.

Desde sua base a QCD mostrou-se complexa, utilizando não apenas da mecânica quântica, mas também da relatividade para a sua construção, além de se apoiar na teoria de grupos para identificar as simetrias, junto das leis de conservação. Por possuir aspectos ainda não conhecidos, como o confinamento de cor, espera-se que seja uma teoria que precise de métodos alternativos para a sua completa compreensão.

Como redes neurais atuam não apenas como aproximadores de funções, mas também como identificadores de padrões, viu-se no problema em questão a oportunidade de utilizar a rede neural como um método alternativo. Contudo, é importante ressaltar que o modelo aqui utilizado não considera os aspectos físicos do problema, apenas os aspectos puramente computacionais.

Portanto, é possível que os resultados aqui apresentados sejam promissores no sentido de iniciar a discussão do uso de aprendizado de máquina em campos inexplorados da Física, servindo como base para futuros trabalhos. Além disso, fica como sugestão de possíveis trabalhos não apenas a utilização de redes neurais que considerem os aspectos físicos do problema (RAISSI; PERDIKARIS; KARNIADAKIS, 2017a,b; RAISSI, 2018), mas também a utilização de redes do tipo aprendizado raso, pois, como visto na seção de resultados e discussão, as melhores redes foram aquelas que possuíam menos camadas.

Referências

- ADHATRAO, Kalpesh et al. Predicting Students' Performance Using ID3 and C4.5 Classification Algorithms. **International Journal of Data Mining & Knowledge Management Process**, Academy e Industry Research Collaboration Center (AIRCC), v. 3, n. 5, p. 39–52, set. 2013. DOI: [10.5121/ijdkp.2013.3504](https://doi.org/10.5121/ijdkp.2013.3504). Disponível em: <<https://doi.org/10.5121/ijdkp.2013.3504>>.
- AGGARWAL, Charu C. **Neural Networks and Deep Learning: A Textbook**. 1st edition: Springer Publishing Company, Incorporated, 2018. ISBN 3319944622.
- AKUTAGAWA, T.; HASHIMOTO, K.; SUMIMOTO, T. Deep learning and AdS/QCD. **Phys. Rev. D**, v. 102, p. 026020, 2 jul. 2020. American Physical Society. DOI: [10.1103/PhysRevD.102.026020](https://doi.org/10.1103/PhysRevD.102.026020). Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevD.102.026020>>.
- BALDI, P.; SADOWSKI, P.; WHITESON, D. Searching for exotic particles in high-energy physics with deep learning. **Nature Communications**, v. 5, n. 1, jul. 2014. Springer Science and Business Media LLC. DOI: [10.1038/ncomms5308](https://doi.org/10.1038/ncomms5308). Disponível em: <<https://doi.org/10.1038/ncomms5308>>.
- BALDI, Pierre; BAUER, Kevin et al. Jet substructure classification in high-energy physics with deep neural networks. **Physical Review D**, v. 93, n. 9, mai. 2016. American Physical Society (APS). DOI: [10.1103/physrevd.93.094034](https://doi.org/10.1103/physrevd.93.094034). Disponível em: <<https://doi.org/10.1103/physrevd.93.094034>>.
- BERTOZZO, Richard Junior. **Aplicação de Machine Learning em Dataset de Consultas Médicas do SUS**. 2019. 100 f. Trabalho de Conclusão de Curso – Universidade Federal de Santa Catarina, Florianópolis.
- BLUM, Avrim; MITCHELL, Tom M. Combining Labeled and Unlabeled Data with Co-Training. In: COLT' 98: THE ELEVENTH ANNUAL CONFERENCE ON COMPUTATIONAL LEARNING THEORY. **COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory**. Madison, Wisconsin, USA: Association for Computing Machinery, 1998. P. 92–100. Disponível em: <<https://dblp.uni-trier.de/db/conf/colt/colt1998.html#BlumM98>>. Acesso em: 25 jan. 2024.
- BRUCE, Rebecca. A Bayesian Approach to Semi-Supervised Learning, fev. 2002.
- CAMPOS, S. D. Chiral symmetry in the confinement phase of QCD. **Modern Physics Letters A**, v. 36, n. 19, p. 2150135, 2021. DOI: [10.1142/S0217732321501352](https://doi.org/10.1142/S0217732321501352). Disponível em: <<https://doi.org/10.1142/S0217732321501352>>.
- CARMO, Carlos Roberto Souza; MELO SILVA, Jéssica Rayse de. Aprendizado de Máquina e Prestação de Serviços de Armazenamento de Dados: Métricas para análise e validação de algoritmos previsores. **GETEC**, v. 12, n. 38, p. 123–144, 2023.
- CHOLLET, François. **Deep Learning with Python**. Manning, 2017.

- COURNAPEAU, D. **Scikit-Learn**. 2007. Disponível em: <<https://scikit-learn.org>>.
- DELANCEY, Evan R. et al. Comparing Deep Learning and Shallow Learning for Large-Scale Wetland Classification in Alberta, Canada. **Remote Sensing**, v. 12, n. 1, 2020. DOI: 10.3390/rs12010002. Disponível em: <<https://www.mdpi.com/2072-4292/12/1/2>>.
- DEUR, A.; BRODSKY, S. J.; DE TÉRAMOND, G. F. The QCD running coupling. **Progr. in Part. and Nucl. Phys.**, v. 90, 2016. DOI: 10.1016/j.pnpnp.2016.04.003. Disponível em: <<https://doi.org/10.1016/j.pnpnp.2016.04.003>>.
- DEUR, Alexandre; BRODSKY, Stanley J; TÉRAMOND, Guy F de. Determination of $\Lambda_{\overline{MS}}$ at five loops from holographic QCD. **Journal of Physics G: Nuclear and Particle Physics**, IOP Publishing, v. 44, n. 10, p. 105005, set. 2017. ISSN 1361-6471. DOI: 10.1088/1361-6471/aa888a. Disponível em: <<http://dx.doi.org/10.1088/1361-6471/aa888a>>.
- DIJKSTRA, Bauke Alfredo. **Reconhecimento de Fonemas Utilizando Redes Neurais Convolucionais para Transcrição Fonética Automática**. 2021. 76 f. Dissertação de Mestrado – Universidade Tecnológica Federal do Paraná, Ponta Grossa.
- DOMINGOS, Pedro. A Few Useful Things to Know about Machine Learning. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 55, n. 10, p. 78–87, out. 2012. ISSN 0001-0782. DOI: 10.1145/2347736.2347755. Disponível em: <<https://doi.org/10.1145/2347736.2347755>>.
- FONTANA, Éliton. **Introdução aos Algoritmos de Aprendizagem Supervisionada**. 2020.
- FUJIMOTO, Y.; FUKUSHIMA, K.; MURASE, K. Mapping neutron star data to the equation of state using the deep neural network. **Phys. Rev. D.**, v. 101, p. 054016–054022, 2020. ISSN 0920-5632. DOI: <https://doi.org/10.1103/PhysRevD.101.054016>.
- GARIA, Gedson; ROMERO, Roseli A. Navegação de Robôs Móveis Utilizando Aprendizado por Reforço e Lógica Fuzzy. **Revista Controle & Automação**, v. 13, n. 3, 2002.
- GÉRON, A. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow**. Alta Books, 2019. ISBN 9788550809021.
- GONÇALVES, André Ricardo. **Redes Neurais Artificiais**. Universidade Estadual de Campinas, 2018.
- GONÇALVES, Jefferson Taneo Shida. **Introdução à QCD: Fundamentos e Aplicações na Física do LHC**. 2017. 61 f. Trabalho de Conclusão de Curso – Instituto de Matemática, Estatística e Física, Universidade Federal do Rio Grande, Rio Grande.
- GRIFFITHS, D. **Introduction to Elementary Particles**. Mörlenbach: Wiler-VCH, 1987.
- GROSS, David J. Twenty Five Years of Asymptotic Freedom. **Nuclear Physics B- Proceedings Supplements**, v. 74, n. 1, 1999. DOI: [https://doi.org/10.1016/S0920-5632\(99\)00208-X](https://doi.org/10.1016/S0920-5632(99)00208-X). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S092056329900208X>>.

- GROUP, Particle Data et al. Review of Particle Physics. **Progress of Theoretical and Experimental Physics**, v. 2020, n. 8, p. 083c01, ago. 2020. ISSN 2050-3911. DOI: [10.1093/ptep/ptaa104](https://doi.org/10.1093/ptep/ptaa104). eprint: <https://academic.oup.com/ptep/article-pdf/2020/8/083C01/34673722/ptaa104.pdf>. Disponível em: <https://doi.org/10.1093/ptep/ptaa104>.
- HARRIS, C. R. et al. Array programming with NumPy. **Nature**, v. 585, n. 7825, p. 357–362, set. 2020. Springer Science and Business Media LLC. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2). Disponível em: <https://doi.org/10.1038/s41586-020-2649-2>.
- HASHIMOTO, K. et al. Deep learning and holographic QCD. **Phys. Rev. D**, v. 98, p. 106014, 10 nov. 2018. American Physical Society. DOI: [10.1103/PhysRevD.98.106014](https://doi.org/10.1103/PhysRevD.98.106014). Disponível em: <https://link.aps.org/doi/10.1103/PhysRevD.98.106014>.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J.H. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. Springer, 2009. (Springer series in statistics). ISBN 9780387848846.
- HAYKIN, Simon. **Redes Neurais: Princípios e Práticas**. São Paulo: Bookman, 2001.
- HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long Short-Term Memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- HORNIK, Kurt; STINCHCOMBE, Maxwell; WHITE, Halbert. Multilayer Feedforward Networks are Universal Approximators. **Neural Networks**, v. 2, p. 359–366, 1989.
- HUNTER, J. D. Matplotlib: A 2D graphics environment. **Computing in Science & Engineering**, v. 9, n. 3, p. 90–95, 2007. IEEE COMPUTER SOC. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- ISLAM, Md et al. Emotion Recognition From EEG Signal Focusing on Deep Learning and Shallow Learning Techniques. **IEEE Access**, v. 9, p. 94601–94624, jun. 2021. DOI: [10.1109/ACCESS.2021.3091487](https://doi.org/10.1109/ACCESS.2021.3091487).
- JAIN, Anil K.; DUBES, Richard C. **Algorithms for Clustering Data**. 1st: Prentice-Hall, 1988.
- LADEIRA, João Damasceno Martins. O algoritmo e o fluxo: Netflix, aprendizado de máquina e algoritmos de recomendações. **Intexto**, n. 47, p. 166–184, ago. 2019.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep Learning. **Nature**, v. 521, p. 436–444, 2015.
- MCKINNEY, W. Data Structures for Statistical Computing in Python. In: WALT, Stéfan van der; MILLMAN, Jarrod (Ed.). **Proceedings of the 9th Python in Science Conference**. 2010. P. 56–61. DOI: [10.25080/Majora-92bf1922-00a](https://doi.org/10.25080/Majora-92bf1922-00a).
- MELLO, Rodrigo Fernandes de; PONTI, Moacir Antonelli. **Machine Learning: A Practical Approach on the Statistical Learning Theory**. 1st: Springer Publishing Company, Incorporated, 2018. ISBN 3319949888.
- MONTEVECHI, André Aoun et al. Machine Learning na Análise do Risco de Crédito: uma Aplicação da Regressão Logística e Algoritmos de Resampling. **XLII Encontro Nacional de Engenharia de Produção**, out. 2022.

- MOREIRA, Marcos Antonio. A Física dos Quarks e a Epistemologia. **Revista Brasileira de Ensino de Física**, v. 29, n. 2, 2007.
- MUELLER, John Paul; MASSARON, Luca. **Aprendizado Profundo para Leigos**. Alta Books, 2020.
- MUHTASIM; MASUD, Md. A. Clustering Countries on COVID-19 Data Among Different Waves Using K-Means Clustering. **Journal of Computer and Communications**, v. 11, p. 1–14, jul. 2023. DOI: [10.4236/jcc.2023.117001](https://doi.org/10.4236/jcc.2023.117001).
- MÜLLER, A.C.; GUIDO, S. **Introduction to Machine Learning with Python: A Guide for Data Scientists**. O'Reilly Media, 2016. ISBN 9781449369897.
- OLAH, Christopher. **Understanding LSTM Networks**. 2015. Disponível em: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- OLDONI DA SILVA, Cristiane; NATTI, Paulo Laerte. Modelo de quarks e sistemas multiquarks. **Revista Brasileira de Ensino de Física**, v. 29, p. 175–187, jan. 2007. DOI: [10.1590/S1806-11172007000200002](https://doi.org/10.1590/S1806-11172007000200002).
- OLVER, Peter J. Emmy Noether's enduring legacy in symmetry. **Symmetry**, v. 29, p. 475–485, 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:52906351>.
- PAIXÃO, Gabriela Miana de Mattos et al. Machine Learning na Medicina: Revisão e Aplicabilidade. **Arquivos Brasileiros de Cardiologia**, Sociedade Brasileira de Cardiologia - SBC, v. 118, n. 1, p. 95–102, jan. 2022.
- PANG, Long-Gang et al. Classify QCD phase transition with deep learning. **Nuclear Physics A**, v. 982, p. 867–870, 2019. The 27th International Conference on Ultrarelativistic Nucleus-Nucleus Collisions: Quark Matter 2018. ISSN 0375-9474. DOI: <https://doi.org/10.1016/j.nuclphysa.2018.10.077>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0375947418303580>.
- RAISSI, Maziar. Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations. **Journal of Machine Learning Research**, v. 19, n. 25, p. 1–24, 2018. Disponível em: <http://jmlr.org/papers/v19/18-046.html>.
- RAISSI, Maziar; PERDIKARIS, Paris; KARNIADAKIS, George E. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. **CoRR**, 2017. abs/1711.10561. arXiv: [1711.10561](https://arxiv.org/abs/1711.10561). Disponível em: <http://arxiv.org/abs/1711.10561>. Acesso em: 4 jul. 2023.
- RAISSI, Maziar; PERDIKARIS, Paris; KARNIADAKIS, George E. Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations. **CoRR**, 2017. abs/1711.10566. arXiv: [1711.10566](https://arxiv.org/abs/1711.10566). Disponível em: <http://arxiv.org/abs/1711.10566>. Acesso em: 4 jul. 2023.
- RAJANAND, Ashish; SINGH, Pradeep. **ErfReLU: Adaptive Activation Function for Deep Neural Network**. 2023. arXiv: [2306.01822](https://arxiv.org/abs/2306.01822) [cs.NE].

- RAUBER, Thomas. **Redes neurais artificiais**. 23 mai. 2014. Disponível em: <https://www.researchgate.net/publication/228686464_Redex_neurais_artificiais>. Acesso em: 4 jul. 2023.
- SAMUEL, Arthur L. Some Studies in Machine Learning Using the Game of Checkers. **IBM Journal of Research and Development**, v. 3, n. 3, p. 210–229, jul. 1959.
- SANTOS, Gustavo Carvalho. **Algoritmos de Machine Learning para Previsão de Ações da B3**. 2020. 94 f. Dissertação de Mestrado – Universidade Federal de Uberlândia - Faculdade de Engenharia Elétrica, Uberlândia.
- SGARBOSA, Pietro; VECHIO, Gustavo. INTELIGÊNCIA ARTIFICIAL E SUAS IMPLICAÇÕES: como os dispositivos inteligentes e assistentes virtuais influenciam o cotidiano das pessoas. **Revista Interface Tecnológica**, v. 17, p. 193–205, dez. 2020. DOI: [10.31510/inf.v17i2.936](https://doi.org/10.31510/inf.v17i2.936).
- SHANAHAN, E. P.; TREWARTHA, D.; DETMOLD, W. Machine learning action parameters in lattice quantum chromodynamics. **Phys. Rev. D**, v. 97, 9 mai. 2018. American Physical Society. DOI: [10.1103/PhysRevD.97.094506](https://doi.org/10.1103/PhysRevD.97.094506). Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevD.97.094506>>.
- SHIRURU, Kuldeep. Neural Network Approach for Processing Substation Alarms. **International Journals of Power Electronics Controllers and Converters**, v. 1, p. 21–28, out. 2015.
- TEAM, The pandas development. **pandas-dev/pandas: Pandas**. Fev. 2020. Zenodo. DOI: [10.5281/zenodo.3509134](https://doi.org/10.5281/zenodo.3509134). Disponível em: <<https://doi.org/10.5281/zenodo.3509134>>.
- TENSORFLOW. 2015. Disponível em: <<https://tensorflow.org>>.
- TETILA, Everton Castelão. **Detecção e classificação de doenças e pragas da soja usando imagens de veículos aéreos não tripulados e técnicas de visão computacional**. 2019. 103 f. Tese de Doutorado – Universidade Católica Dom Bosco, Campo Grande.
- THOMSON, M. **Modern Particle Physics**. New York: Cambridge University Press, 2013.