

CENTRO UNIVERSITÁRIO FEI
GABRIELA MOLINA CIOCCI

**USO DE REDES NEURAIIS PARA ANÁLISE DE DADOS NAS DISCIPLINAS
DE LABORATÓRIO DE FÍSICA**

São Bernardo do Campo

2024

GABRIELA MOLINA CIOCCI

**USO DE REDES NEURAIS PARA ANÁLISE DE DADOS NAS DISCIPLINAS
DE LABORATÓRIO DE FÍSICA**

Relatório Parcial de Iniciação
Didática apresentado ao Centro
Universitário FEI, como parte
dos requisitos do Programa
PIBIC-FEI. Orientado pela Prof.
Eliane de Fátima Chinaglia

São Bernardo do Campo

2024

RESUMO

Há muito tempo a formação em engenharia é desenvolvida apenas em sua parte técnica. Porém, o alto avanço da tecnologia implica a necessidade do aluno se tornar mais ativo em seus estudos. Analisar simulações e comparar com dados experimentais são ótimas opções para tornar o estudante protagonista de seus estudos. Com o auxílio de ferramentas e algoritmos, os alunos poderão trabalhar de forma mais prática e eficiente alguns tópicos que demandam mais tempo para compreensão. Utilizando Python como linguagem de programação, Scikit-Learn como biblioteca de Rede Neural e Matplotlib como biblioteca responsável por esboçar gráficos, o objetivo é criar um software intuitivo que auxilie o aluno de engenharia a entender melhor os conceitos e dados. Dados experimentais foram obtidos com um trilho de ar, responsável por minimizar as forças de atrito do movimento dois corpos de massas diferentes acoplados por um fio em uma polia, juntamente com sensores responsáveis por captarem a variação da velocidade e tempo, com o objetivo de treinar uma Rede Neural.

Palavras-chave: Engenharia. Metodologias ativas. Rede Neural. Python. Machine Learning. Scikit-Learn. Física Básica.

LISTA DE ILUSTRAÇÕES

Figura 1 - Exemplo de uma rede poligonal de Thiessen	12
Figura 2 - Neurônio Cerebral	14
Figura 3 - Modelo de Neurônio Artificial.....	15
Figura 4 - Função Sigmoide.....	16
Figura 5 - Exemplo de rede neural artificial.....	17
Figura 6- Diagrama de forças do sistema	17
Figura 7 - Arranjo experimental do trilho de ar.....	19
Figura 8 - Programa de automação em Python.....	21
Figura 9 - Método de Elbow dos dados do projeto	24
Figura 10 - Análise de clusters obtidos	25
Figura 11 - Aproximação da rede poligonal de Thiessen do experimento	26
Figura 12 - Box Plot da aceleração em função de massa m_2	27
Figura 13 - Cronograma anual do projeto	29

LISTA DE TABELAS

Tabela 1 - Variação das massas utilizadas no experimento	20
Tabela 2 - Resultado do agrupamento pelo k-Means.....	24

SUMÁRIO

1 INTRODUÇÃO	6
2 REVISÃO BIBLIOGRÁFICA	8
2.1 Análise de agrupamento não hierárquico	8
2.1.1 Modelo geral de agrupamento	9
2.1.2 Método k-Means	10
2.1.2.1 História do método k-Means	11
2.1.3 Polígonos de Thiessen (Voronoi)	11
2.2 Rede Neural	12
2.2.1 Neurônio Cerebral	13
2.2.2 Neurônio Artificial	14
2.3 Cinemática e Dinâmica de dois corpos acoplados	17
3 METODOLOGIA	19
3.1 Aquisição de dados	20
3.1.1 Definição dos parâmetros	20
3.2 Desenvolvimento de programas para a aquisição de dados	21
3.3 Utilização de bibliotecas	22
4 RESULTADOS OBTIDOS	23
4.1 Dados adquiridos	23
4.2 Método de Elbow	23
4.3 Formação dos Clusters	24
4.4 Análise de dados utilizando box plot	27
5 CONCLUSÃO PARCIAL E PRÓXIMAS ETAPAS	28
6 CRONOGRAMA	29
REFERÊNCIAS	30

1 INTRODUÇÃO

Um dos aspectos fascinantes do campo da Inteligência Artificial (IA) é que sua definição é complexa, e para garantir uma compreensão adequada do que é artificial, é necessário primeiramente obter uma compreensão do que é inteligência. Inteligência é a capacidade de processar informações, raciocinar, resolver problemas, aprender com experiências e adaptar-se ao ambiente. O que pode definir o “artificial” na IA é o fato de sua criação ser um produto da engenhosidade humana, no entanto, apresentando resultados e decisões de um ponto de vista evolutivo em comparação as decisões tomadas pelos humanos. Em outras palavras, coisas que são artificialmente inteligentes tem a capacidade de processar informação, aprender e tomar decisões similarmente a organismos vivos, porém com propriedades especiais como agilidade e trabalhos com uma escala que apenas a máquina consegue analisar. (FETZER, 1990).

O Machine Learning (ML) é um subconjunto da IA que permite que sistemas aprendam e melhorem de forma autônoma, a partir de dados. Ao alimentar algoritmos de aprendizado com grandes conjuntos de dados, são construídos modelos capazes de identificar padrões e fazer previsões (ZHOU, 2021). Essa capacidade de aprender a partir da experiência torna o ML uma ferramenta poderosa em diversas áreas, desde a análise de dados até o desenvolvimento de sistemas autônomos.

Observa-se hoje um crescimento acentuado do uso de algoritmos de IA e ML não apenas no cotidiano, mas também no mercado de trabalho. Assim, a formação dos alunos em um curso de Engenharia, torna-se essencial que os futuros engenheiros adquiram, pelo menos, uma compreensão básica desses algoritmos. Tendo ciência dessa importância, o Laboratório de Física irá contribuir a fim de oferecer aos futuros engenheiros conceitos básicos de máquinas inteligentes (MEI/CNI, 2018).

O objetivo deste projeto é apresentar a um aluno de engenharia os conceitos básicos de uma IA interligados aos conceitos da Segunda Lei de Newton. Além disso, este projeto também visa aproveitar a oportunidade de

coletar e analisar dados durante as aulas de laboratório para proporcionar aos alunos ingressantes a interação com algoritmos básicos de ML. Com isso, o aluno poderá aprender conhecimentos de física juntamente com os conceitos de IA e programação, destacando a importância dessa habilidade em sua formação (MITCHELL, 1997).

Após o processo de treinamento da Rede Neural, o aluno irá realizar medidas de aceleração no laboratório em função da massa de um corpo e do ângulo de inclinação do trilho e a rede neural utilizará essas informações para classificar a massa do outro corpo.

Neste projeto serão explorados algoritmos de ML tanto não supervisionados (agrupamento) quanto supervisionados (Rede Neural) para analisar dados obtidos em experimentos realizados pelos alunos no laboratório de física. A análise de agrupamento não hierárquico, também conhecido como partição, é uma técnica de aprendizado não supervisionado que busca dividir um conjunto de dados em um número pré-definido de grupos (clusters) de forma que os objetos dentro de cada grupo sejam mais semelhantes entre si do que com os objetos de outros grupos (FÁVERO, 2022). Redes neurais artificiais são modelos computacionais inspirados no sistema nervoso humano, capazes de aprender e tomar decisões complexas a partir de dados (GIORDANI, 2020).

2 REVISÃO BIBLIOGRÁFICA

Visando que o objetivo do projeto consiste em enriquecer o ensino introdutório de Física através da IA, é essencial refinar os conhecimentos sobre os temas trabalhados neste projeto. O tópico selecionado para a criação de uma simulação foi a cinemática e a dinâmica das partículas.

2.1 ANÁLISE DE AGRUPAMENTO NÃO-HIERÁRQUICO

A análise de agrupamento não hierárquico, também conhecido como partição, é uma técnica de aprendizado não supervisionado que busca dividir um conjunto de dados em um número pré-definido de grupos (clusters) de forma que os objetos dentro de cada grupo sejam mais semelhantes entre si do que com os objetos de outros grupos (FAVERO, 2022).

Um dos algoritmos de agrupamento não hierárquicos mais famoso é o k-Means. Os protótipos k-Means, é uma técnica de quantização vetorial que divide um conjunto de N observações em K clusters, em que cada observação pertence ao cluster cuja sua média (também chamada de centroide) seja a mais próxima. (GIORDANI, 2020).

Para criar um agrupamento, é necessário definir a quantidade inicial de grupos e seus respectivos centroides. O objetivo desse passo é dividir os itens do banco de dados em K grupos, de modo que todos os itens em cada grupo estejam mais próximos entre si se comparados a qualquer outro item pertencente a um grupo diferente. Para isso, os itens precisam ser alocados arbitrariamente nos grupos para que assim possa ser calculado os centroides através da média aritmética do grupo em relação a cada item (FAVERO, 2022).

Deve-se selecionar determinado item que se encontra mais próximo de um centroide e realocá-lo nesse grupo, fazendo assim com que outro grupo perca esse item e, portanto, recalcular os centroides do grupo que recebeu e do grupo que perdeu o item. Esse passo deve ser feito até que não seja mais possível realocar nenhum item por proximidade (FAVERO, 2022).

2.1.1 Modelo geral de agrupamento

Para realização de um agrupamento não hierárquico, será necessário definir dois conjuntos d-dimensionais de vetores reais. Um deles será um conjunto de N pontos de dados $X = \{x_1, \dots, x_n\}$ que contém os dados a serem agrupados. O outro será um conjunto de K centros $C = \{c_1, \dots, c_k\}$ que contém a quantidade de cluster que o algoritmo deverá refinar. (HAMERLY, 2002).

Uma função de associação $M(c_j|x_i)$ define a proporção de pontos de dados x_i que pertence ao centro c_j e suponha que exista restrições que limitam os pontos entre $M(c_j|x_i) \leq 1$ e $\sum_{j=1}^k M(c_j|x_i) = 1$. Existem dois principais tipos de função de associação, são elas:

- Suave: Nessa função, um ponto pode pertencer a mais de um grupo, pois atribui para cada ponto uma probabilidade de pertencer a um cluster ($0 \leq M(c_j|x_i) \leq 1$).
- Rígida: Nessa função, um ponto pertence exclusivamente a um único grupo. Ou seja, se um ponto é atribuído a um grupo, ele não pode pertencer a nenhum outro ($M(c_j|x_i) \in \{0,1\}$). É como colocar cada ponto em uma caixa separada e distinta.

Uma função de peso $w(x_i)$ define o quanto de influência o banco de dados de pontos tem para computar novamente os parâmetros centrais na próxima iteração, com restrição $w(x_i) > 0$. Com isso, podemos obter um modelo geral de clusterização iterativa baseada em centroides (HAMERLY, 2002):

1. Inicialize o algoritmo com a quantidade de centroides desejada;
2. Para cada ponto no banco de dados X, calcular sua associação $M(c_j|x_i)$ em cada centroide c_j e seu peso $w(x_i)$;
3. Para cada centroide c_j , recalcule todos os pontos do banco de dados X de acordo com suas associações e pesos, descrita na equação 1:

$$c_j = \frac{\sum_{i=1}^n m(c_j|x_i)w(x_i)x_i}{\sum_{i=1}^n m(c_j|x_i)w(x_i)} \quad (1)$$

4. Repita as etapas 2 e 3 até que o algoritmo tenha convergência.

A função rígida torna possível muitas otimizações computacionais que não afetam a precisão do algoritmo. O k-Means utiliza dessa função para realizar seus cálculos (HAMERLY, 2002).

A seguir, será explicado o que é o método k-Means e como é seu funcionamento.

2.1.2 Método k-Means

O algoritmo k-Means separa os dados em k grupos, tendo como solução um conjunto de k grupos, cada um localizado no centroide dos dados nos quais é o centro mais próximo. Isso resulta em um particionamento do espaço de dados em células Voronoi, que será explicado após esse tópico. O agrupamento de k-Means minimiza as variâncias dentro do agrupamento (distâncias euclidianas ao quadrado), mas não as distâncias euclidianas regulares. A equação 2 informa qual equação o método utiliza em seu algoritmo (HAMERLY, 2002).

$$KM(X, C) = \sum_{l=1}^n \min_{j \in \{1, \dots, k\}} \|x_l - c_j\|^2 \quad (2)$$

n : tamanho do vetor d-dimensional $X = \{x_1, \dots, x_n\}$

x_i : ponto localizado na posição i do vetor X

c_j : ponto localizado na posição j do vetor $C = \{c_1, \dots, c_k\}$

k : tamanho do vetor d-dimensional C

2.1.2.1 História do método k-Means

O termo k-Means foi utilizado pela primeira vez por James MacQueen em 1967. Porém, o algoritmo foi proposto pela primeira vez por Stuart P. Lloyd em 1957 como uma técnica para Quantização por Mínimos Quadrados (LSQ) em modulação de código de pulso, onde visava minimizar a distorção entre um sinal original e sua representação quantizada. Em termos simples, ela divide o intervalo de valores possíveis de um sinal em um número finito de níveis de quantização e atribui a cada amostra do sinal o nível mais próximo (STUART, 1957).

Dado um conjunto de dados e um número fixo de níveis de quantização, o objetivo da LSQ é encontrar os valores ótimos para esses níveis de modo a minimizar o erro quadrático médio entre os dados originais e suas representações quantizadas (STUART, 1957). Por isso, o método k-Means também pode ser referenciado como “Lloyd-Forgy algorithm”.

Ambos os problemas são essencialmente os mesmos, com diferentes interpretações:

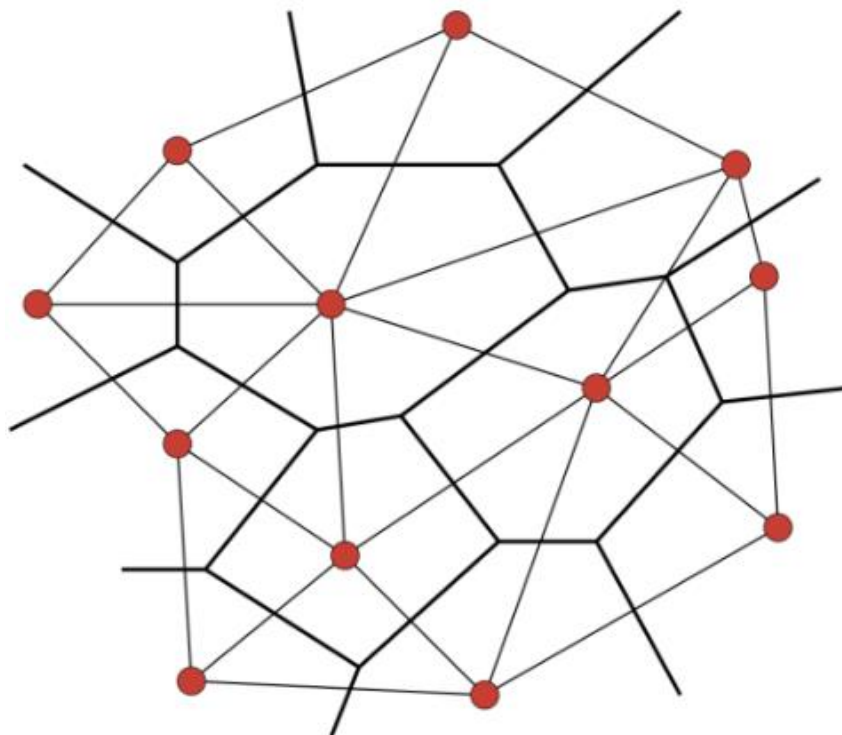
- LSQ: O objetivo é minimizar o erro de quantização entre o sinal original e sua representação quantizada;
- k-Means: O objetivo é particionar um conjunto de dados em k clusters de modo que a soma das distâncias quadráticas entre cada ponto de dados e o centroide de seu cluster seja minimizada.

2.1.3 Polígonos de Thiessen (Voronoi)

Os polígonos de Thiessen (também chamados de Dirichlet ou Voronoi) é uma partição de um plano em regiões próximas a cada um de um determinado conjunto de objetos. Se os dados estiverem em uma grade quadrada regular, então os polígonos de Thiessen são todos iguais (BURROUGH, 2015). Se os dados forem espaçados irregularmente, então o resultado obtido será uma rede

irregular de polígonos, demonstrada na figura 1, onde os círculos indicam os centroides.

Figura 1- Exemplo de uma rede poligonal de Thiessen



Fonte: Burrough PA, McDonnell R, Lloyd C, 2015. “Principles of Geographical Information Systems.” 3rd ed. p-16.

2.2 REDE NEURAL

Em 1943, o primeiro modelo matemático de um neurônio biológico foi apresentado por McCulloch e Pitts. Esse modelo considerou o neurônio como uma unidade de processamento de informações binárias, contendo várias entradas binárias e uma única saída binária (MCCULLOCH, 1943).

Em 1958, Frank Rosenblatt cria o Perceptron, um dos primeiros modelos de aprendizado de máquina capazes de aprender a partir de dados. O Perceptron funciona calculando uma combinação linear das entradas, ponderadas por pesos específicos. Se essa soma excede um determinado valor

limiar, o Perceptron gera uma saída positiva; caso contrário, gera uma saída negativa (ROSENBLATT, 1958).

Uma Rede Neural Artificial (RNA) é uma imitação do cérebro humano. O cérebro humano tem capacidades incríveis como aprender coisas novas, adaptar-se a mudanças, interpretar informações que ficaram incompletas e conseguir obter suas próprias conclusões, reconhecer toques, cheiros ou vozes, ter a capacidade de imaginar, sonhar, aprender, além de ser responsável por desempenhar diversas funções importantes para o funcionamento do corpo humano (KUKREJA, 2016).

O cérebro é um órgão extremamente complexo no qual nenhuma máquina tem capacidade de realizar todas as suas capacidades até o momento.

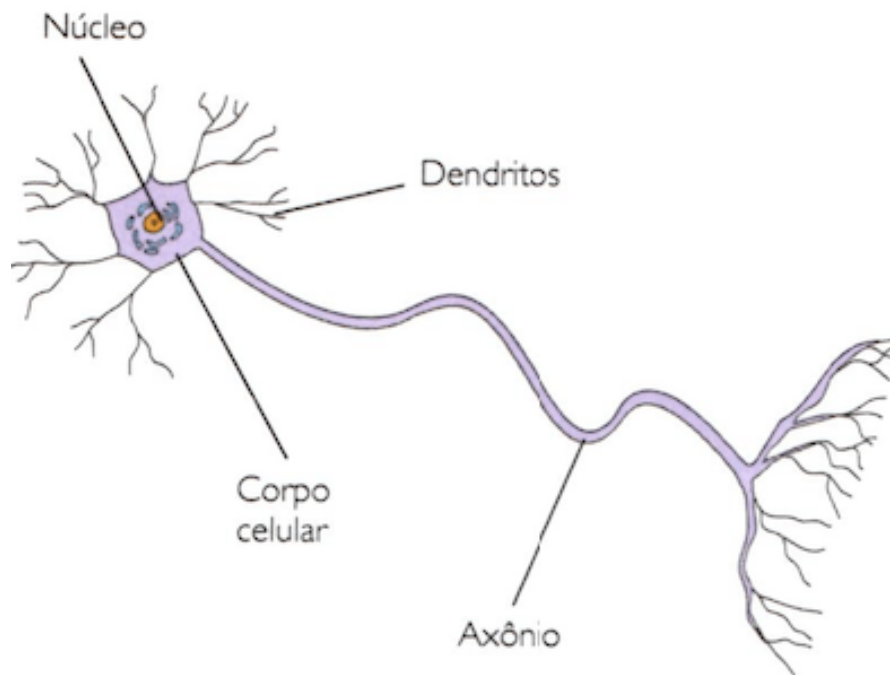
2.2.1 Neurônio Cerebral

O cérebro é composto de neurônios, que são células nervosas altamente estimuláveis, responsáveis por processarem e transmitir informações através de sinais eletroquímicos (MOREIRA, 2013). Em sua maioria, os neurônios possuem três partes principais:

- **Corpo celular:** É a parte central da célula, onde fica o núcleo e a maior parte das organelas (MOREIRA, 2013);
- **Dendritos:** São ramificações curtas que recebem os sinais de outros neurônios. São como “antenas” (MOREIRA, 2013);
- **Axônio:** É um prolongamento longo e fino que conduz os sinais para outros neurônios, músculos ou glândulas. É como um "cabo" que transmite a informação (MOREIRA, 2013).

A figura 2 demonstra a estrutura de um neurônio cerebral.

Figura 2 - Neurônio Cerebral



Fonte: Bardine, Renan. Cola da Web, 2000-2024. “Neurônios”.

Quando a terminação do axônio de um neurônio estabelece ligações com os dendritos de um outro neurônio, esse neurônio transmite seu próprio sinal por meio do axônio utilizando sinapse, que permite que o impulso nervoso seja conduzido de um neurônio para o seguinte. O sinal enviado a outros neurônios por meio de sinapses o dispara, e faz com que o processo continue (MOREIRA, 2013).

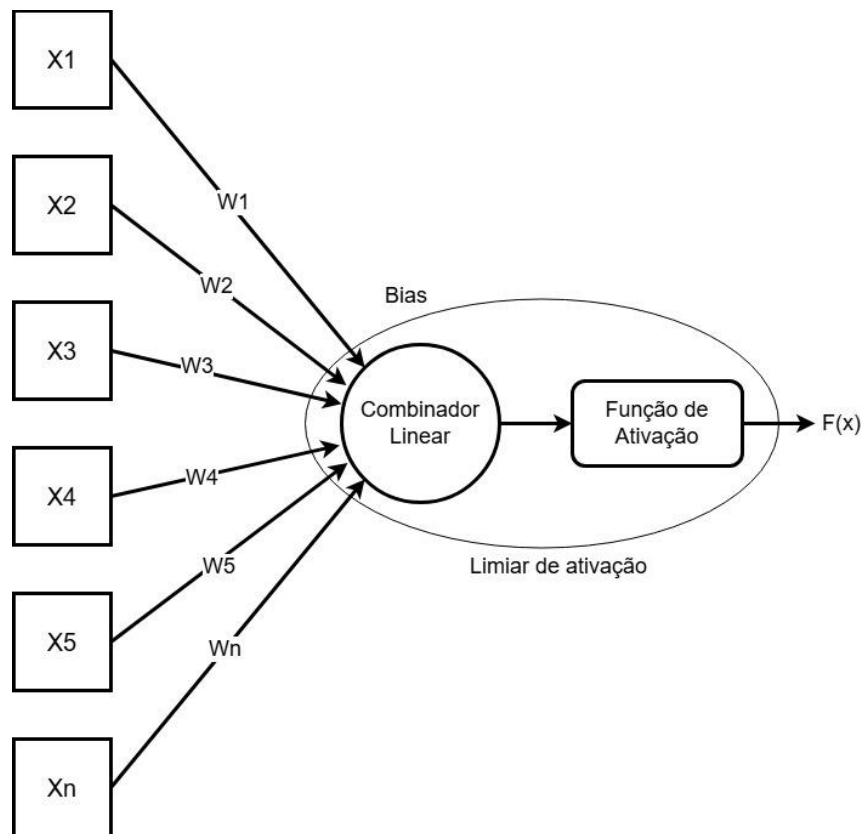
A transmissão de um impulso nervoso segue o princípio do “tudo ou nada”, significa que um estímulo precisa ter uma intensidade mínima para desencadear um potencial de ação, chamado de estímulo limiar. Ao atingir o limiar, a intensidade do estímulo não altera a força do potencial de ação, mas influencia na quantidade de impulsos por segundo (MOREIRA, 2013).

2.2.2 Neurônio Artificial

Um neurônio artificial tenta replicar o comportamento dos neurônios naturais e consiste em unidades de processamento chamadas de neurônios. Ele

é composto por entradas (dendritos) e em uma única saída (axônio). O neurônio possui uma função de ativação que determina sua ativação. A figura 3 ilustra o modelo de um neurônio artificial.

Figura 3 - Modelo de Neurônio Artificial



Fonte: Gabriela Molina Ciocci

No qual $X = \{x_1, \dots, x_n\}$ são as entradas, $W = \{w_1, \dots, w_n\}$, que são a conexão e $F(x)$ é a saída.

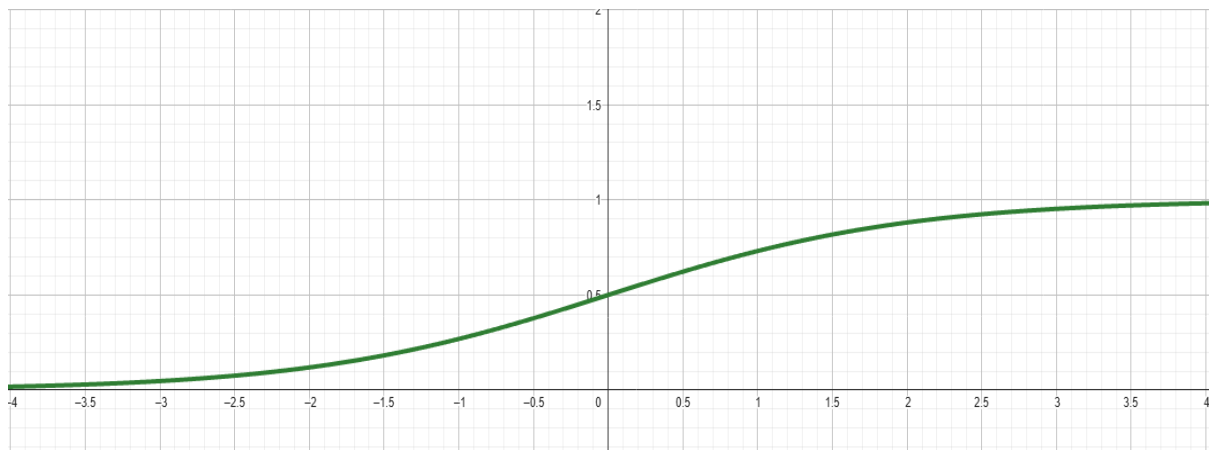
A principal função do peso é dar importância às características que mais contribuem para o aprendizado. Ao multiplicar os valores de entrada pelos pesos, o modelo "decide" quais informações são mais relevantes para a tarefa em questão. Por exemplo, em um modelo de análise de sentimentos, uma palavra negativa pode receber um peso maior do que palavras neutras, influenciando significativamente a classificação final (KUKREJA, 2016).

Juntamente com as entradas, é adicionado um bias, seu papel é mudar o valor produzido pela função de ativação (KUKREJA, 2016).

A função de ativação, por sua vez, introduz não linearidade dos Perceptrons, permitindo que eles capturem padrões mais complexos nos dados. Sem ela, a saída do modelo seria apenas uma combinação linear das entradas. Existem várias funções de ativação, mas a mais comum é a sigmoide, dada pela equação (3) e ilustrada pela figura 4 (KUKREJA, 2016).

$$F(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

Figura 4 - Função Sigmoide



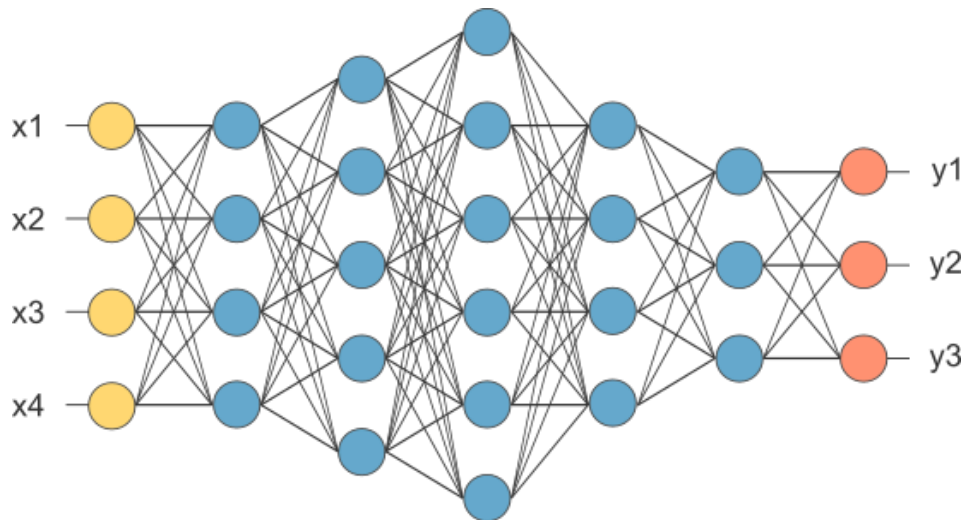
Fonte: Gabriela Molina Ciocci

A função de ativação sigmoide é muito utilizada pois seus limites são entre 0 e 1. A variável x é a soma das entradas multiplicada pelos pesos entre uma camada até a próxima. A interconexão desses neurônios forma uma Rede Neural (KUKREJA, 2016).

Os dados de entrada são alimentados na primeira camada, e a informação flui através das camadas ocultas, que são um conjunto de neurônios. Pode haver várias camadas ocultas.

Esse processo perdura até a camada de saída, onde é gerada a resposta final. Pode haver mais de uma saída, mas não é comum, e seu resultado varia entre 0 e 1 (ZHANG, 2018). Um exemplo de Rede Neural é apresentado na figura 5.

Figura 5 - Exemplo de rede neural artificial

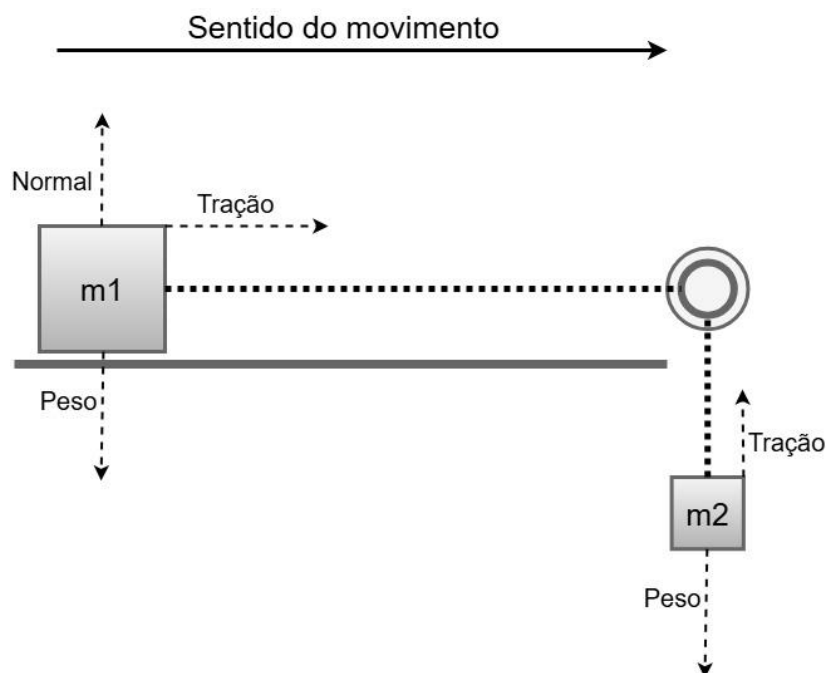


Fonte: Coelho, Matheus. Laboratório iMobilis, 2017. “Fundamentos de Redes Neurais”.

2.3 CINEMÁTICA E DINÂMICA DE DOIS CORPOS ACOPLADOS

Os tópicos de física básica abordados nesse projeto são a cinemática e a dinâmica de partículas. A figura 6 ilustra o sistema experimental utilizado.

Figura 6- Diagrama de forças do sistema



Fonte: Gabriela Molina Ciocci

Nesse projeto, foi utilizado um trilho de ar responsável por minimizar a força de atrito, tornando-a praticamente nula. Por esse motivo, a força de atrito não está presente no diagrama.

Os dados extraídos são numéricos e foram classificados de tal forma:

m_1 : massa do corpo posicionado em um plano horizontal sem atrito;

m_2 : massa do corpo suspenso;

Para visar os resultados experimentais, é de extrema importância analisar a parte teoria do problema, por isso, calcular a aceleração teórica auxilia a analisar se os resultados obtidos estão sendo próximos aos resultados esperados.

Para determinar a aceleração teórica de um corpo, recorreremos à Segunda Lei de Newton, um dos pilares da dinâmica. Essa lei estabelece uma relação fundamental entre três grandezas físicas: a força resultante (F) aplicada a um corpo, de massa (m) e a aceleração (a) que ele adquire. Neste caso, o módulo da aceleração do sistema é dado pela equação (4).

$$\vec{F} = m \times \vec{a} \quad (4)$$

\vec{F} : resultante que atua sobre o corpo (N);

m : massa do corpo (kg);

\vec{a} : aceleração do corpo (m/s^2).

Ao mencionar a Segunda Lei de Newton, fica claro que o uso desta lei no projeto é essencial, porém é necessário levar em consideração que temos um movimento acoplado de dois corpos sem a presença do atrito. Dessa forma, a aceleração é dada pela equação (5).

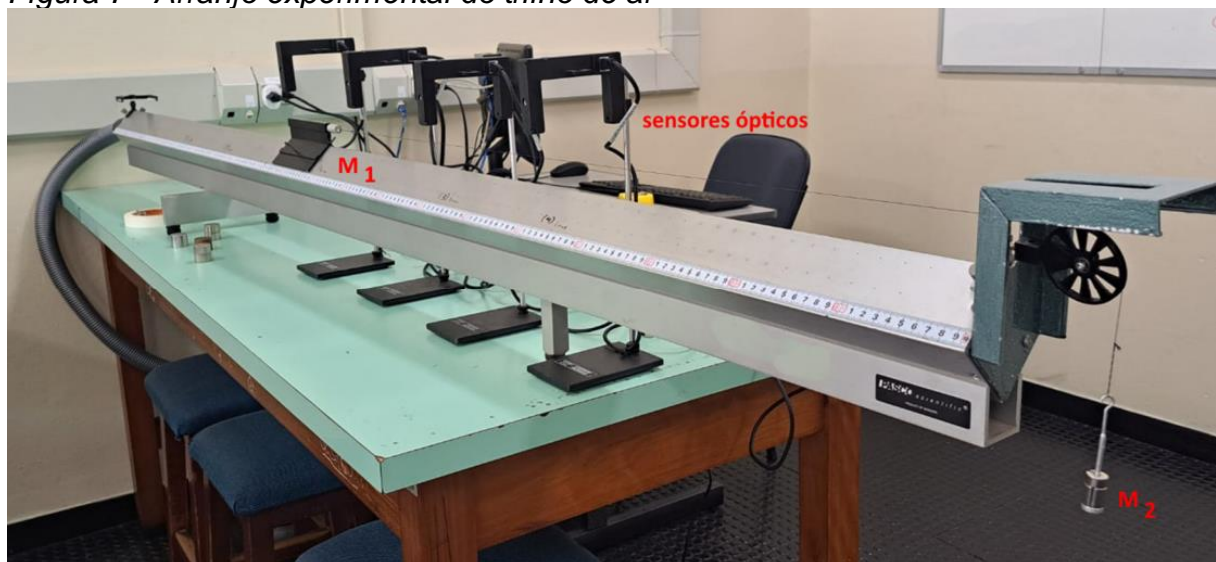
$$a = \frac{m_2 \times g}{m_1 + m_2} \quad (5)$$

Após feito as devidas alterações na fórmula e manter a aceleração devidamente isolada para fins de calculá-la, é possível comparar o resultado obtido através do experimento com a aceleração esperada.

3 METODOLOGIA

A primeira etapa para o desenvolvimento desse projeto consistiu na aquisição de dados para obter um banco de dados eficiente e confiável, com o intuito de treinar a IA. Foi utilizado um trilho de ar a fim de minimizar as forças de atrito no movimento de dois corpos de massas diferentes e acoplados por um fio. Sensores ópticos posicionados estrategicamente permitiram a medida da velocidade de um dos corpos (m_1). Com esse arranjo experimental, foi possível determinar a aceleração do sistema a partir das medidas de velocidade em função do tempo. O trilho pode ser horizontal ou inclinado, formando um ângulo. Nesta etapa, o trilho de ar foi mantido a 0° , ou seja, horizontalmente. A figura 7 apresenta o arranjo experimental do projeto.

Figura 7 - Arranjo experimental do trilho de ar



Fonte: Gabriela Molina Ciocci, Bruno Arthur Basso Silva

Na tabela 1, está apresentada a variação de valores de m_1 e m_2 utilizada para a aquisição de dados do projeto.

Tabela 1 - Variação das massas utilizadas no experimento

m_1	m_2
0.209	0.004 a 0.013
0.309	0.014 a 0.019
0.409	0.02 a 0.03

Fonte: Gabriela Molina Ciocci

3.1 AQUISIÇÃO DOS DADOS

Para a aquisição de dados foi utilizado o programa Data Studio, que fornece a velocidade e tempo de forma extremamente precisa. Ao realizar o experimento, com o movimento de um dos corpos no trilho de ar, o software registra meticulosamente em intervalos regulares de tempo e com o auxílio de quatro sensores posicionados simetricamente. Em seguida, os dados coletados foram armazenados uma planilha no Excel.

3.1.1 Definição dos parâmetros

Os dados de entrada para treinar uma IA podem ser numéricos ou categóricos, desde estejam formatados em um padrão para que a Rede Neural possa classificar os dados de forma eficaz.

Para realização do projeto, foi necessário identificar quais parâmetros são indispensáveis para que resultado obtido seja conforme o esperado. Após realizar essa análise, as variáveis escolhidas para melhor execução do algoritmo de classificação foram:

- m_1 : massa do corpo posicionado no trilho de ar
- \vec{a} : aceleração do corpo (m/s^2)

Foram escolhidas essas variáveis pois o principal objetivo do projeto consiste em treinar a Rede Neural para poder classificar qual grupo de massas

m_2 o experimento se encontra. Ainda não foi adicionado nenhum dado que tenha variação de ângulo, por esse motivo a variável não foi mencionada, mas futuramente ela será acrescentada tendo em vista que sua variação ocasionará mudanças na eficácia da Rede Neural.

3.2 DESENVOLVIMENTO DE PROGRAMAS PARA A AQUISIÇÃO

Para otimizar e facilitar o trabalho experimental, foi desenvolvido um programa que automatiza a adição dos dados obtidos pelo Data Studio no banco de dados no Excel utilizando a linguagem de programação Python e o auxílio da biblioteca Pandas. A figura 8 mostra o terminal integrado do software VSCode.

Figura 8 - Programa de automação em Python

```

-----> DADO 1 <-----

----- TEMPO -----

Tempo 1: 2.0863
Tempo 2: 2.4774
Tempo 3: 2.7567
Tempo 4: 2.9813

----- VELOCIDADE -----

Velocidade 1: 36
Velocidade 2: 63
Velocidade 3: 82
Velocidade 4: 98

Aceleração calculada: 69.27374301675978

Dados importados para o arquivo Excel: dados.xlsx

```

Fonte: Gabriela Molina Ciocci, Bruno Arthur Basso Silva

3.3 UTILIZAÇÃO DE BIBLIOTECAS

Para realização do projeto, foi preciso utilizar algumas bibliotecas em Python para o desenvolvimento da IA e para análise de gráficos (PYTHON, 2001). São elas:

- Scikit-Learn: também conhecido como sklearn, é uma das bibliotecas mais populares em Python para aprendizado de máquina. Ela oferece uma ampla variedade de algoritmos de aprendizado de máquina, tanto para tarefas supervisionadas quanto não supervisionadas (SCIKIT-LEARN, 2007).
- Pandas: é amplamente utilizada para manipulação e análise de dados. Ela oferece estruturas de dados poderosas, como o Data Frame, que permite armazenar e manipular conjuntos de dados de forma eficiente. Além disso, é possível realizar operações de limpeza, transformação e análise estatística em dados tabulares de maneira intuitiva e eficaz (PANDAS, 2008).
- Matplotlib.pyplot: é uma sub-biblioteca do Matplotlib, que fornece uma interface para criar gráficos de alta qualidade em Python. Com ele, é possível criar uma ampla variedade de gráficos, como gráficos de linha, histogramas, gráficos de dispersão e muito mais (MATPLOTLIB, 1991).
- Seaborn: é uma biblioteca de visualização de dados que fornece uma interface de alto nível para criar gráficos estatísticos atraentes e informativos. E simplifica a criação de gráficos complexos, como diagramas de dispersão com regressão linear, box plots e mapas de calor (SEABORN, 2012).
- SciPy: é uma biblioteca usada para computação científica e técnica. Ela é construída sobre o NumPy e fornece funcionalidades adicionais para operações matemáticas, otimização, processamento de sinais, álgebra linear, integração numérica, interpolação, estatísticas, entre outros. (SCIPY, 2008).

4 RESULTADOS OBTIDOS

Após uma imersão completa no projeto, compreendendo suas nuances e fundamentos, é necessário sintetizar todos os resultados obtidos até o momento. Essa etapa é essencial para identificar tanto os aspectos positivos que impulsionam o projeto quanto os desafios que ainda precisam ser superados. Com base nessa análise criteriosa, é possível traçar um caminho mais eficiente para a continuidade do projeto, maximizando ainda mais suas chances de sucesso.

4.1 DADOS ADQUIRIDOS

Os resultados experimentais estão alinhados com as previsões teóricas, indicando que a coleta de dados está sendo realizada de forma eficiente conforme o planejado. Essa precisão na aquisição de dados é fundamental para o sucesso do treinamento da Rede Neural, pois garante que o modelo seja exposto a dados de alta qualidade e representativos do fenômeno em estudo contribuindo significativamente para a construção de um banco de dados robusto. Para treinar a Rede Neural, até o momento, foram adquiridos 256 dados, tendo uma variedade grande de valores para m_2 .

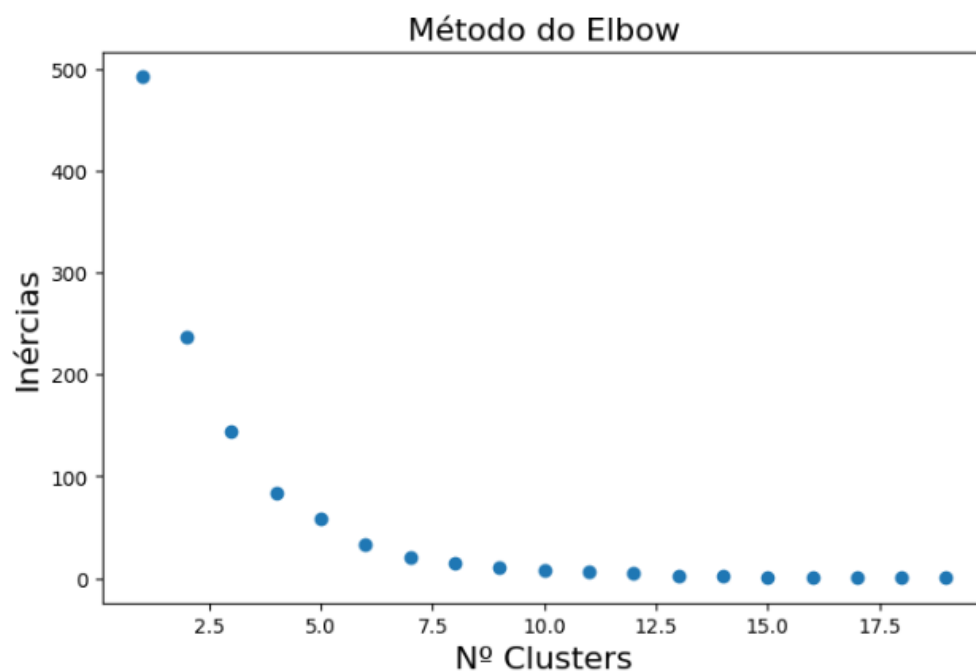
4.2 MÉTODO DE ELBOW

Para determinar a quantidade ideal de clusters em um conjunto de dados, foi utilizado o método de Elbow.

O método de Elbow é um método utilizado para descobrir a quantidade ideal de clusters em um conjunto de dados. O método é chamado dessa forma pois ele sempre está em busca do ponto de inflexão na curva, assemelhando-se ao formato de um cotovelo. Esse método inicia com um único cluster e vai adicionando mais clusters gradualmente. A cada novo cluster, a qualidade da classificação é avaliada. Quando a melhora na classificação se torna muito pequena, ou seja, quando os ganhos em precisão se tornam insignificantes, o processo para. Nesse ponto, a divisão em clusters adicionais não traz mais

benefícios significativos (YELLOWBRICK, 2016). A figura 9 apresenta o gráfico do método utilizado com o banco de dados adquirido através desse projeto.

Figura 9- Método de Elbow dos dados do projeto



Fonte: Gabriela Molina Ciocci

4.3 FORMAÇÃO DOS CLUSTERS

Após a análise do gráfico, foi observado que a quantidade mais eficaz para o projeto até o momento seria de 5 clusters. A tabela 2 apresenta os resultados de agrupamento do k-Means das massas m_2 .

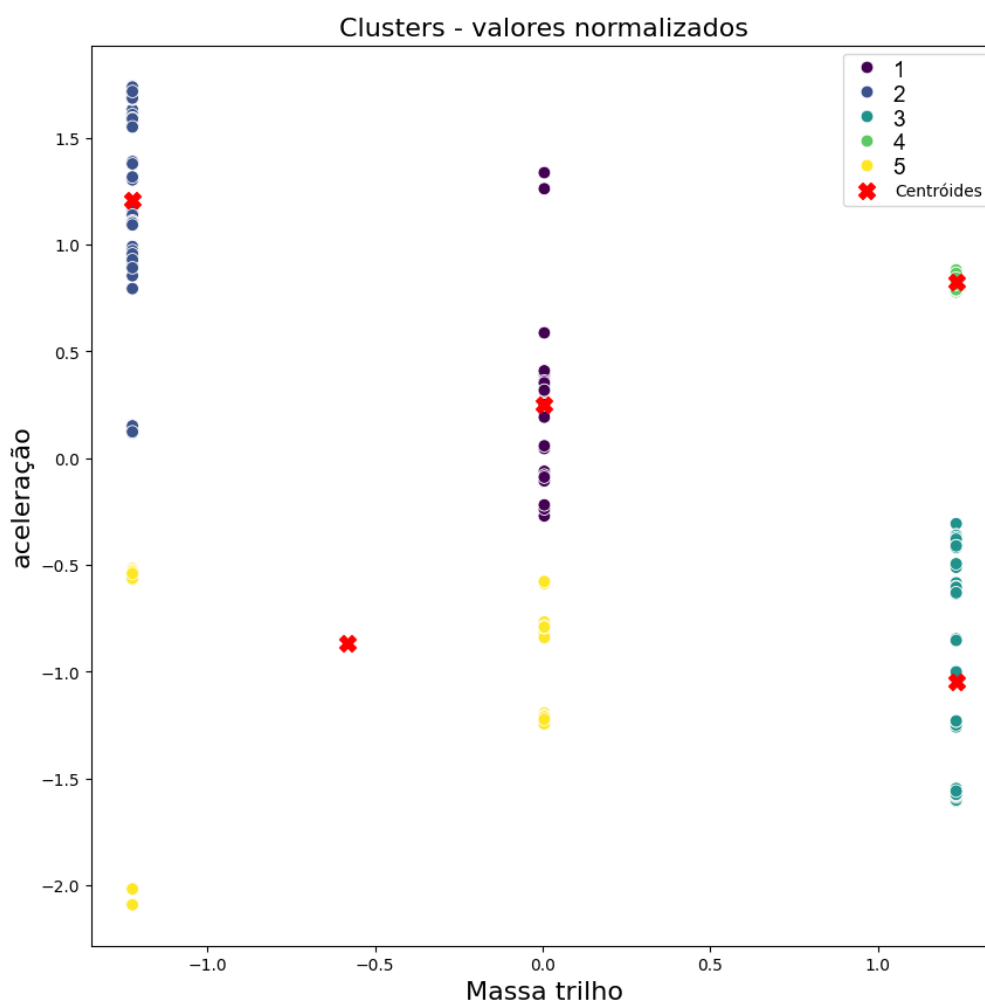
Tabela 2- Resultado do agrupamento pelo k-Means

Cluster	Quantidade de itens por Cluster	Valor mínimo de m_2	Valor máximo de m_2
1	59	0.015	0.021
2	60	0.013	0.020
3	61	0.009	0.021
4	20	0.031	0.031
5	46	0.004	0.015

Fonte: Gabriela Molina Ciocci

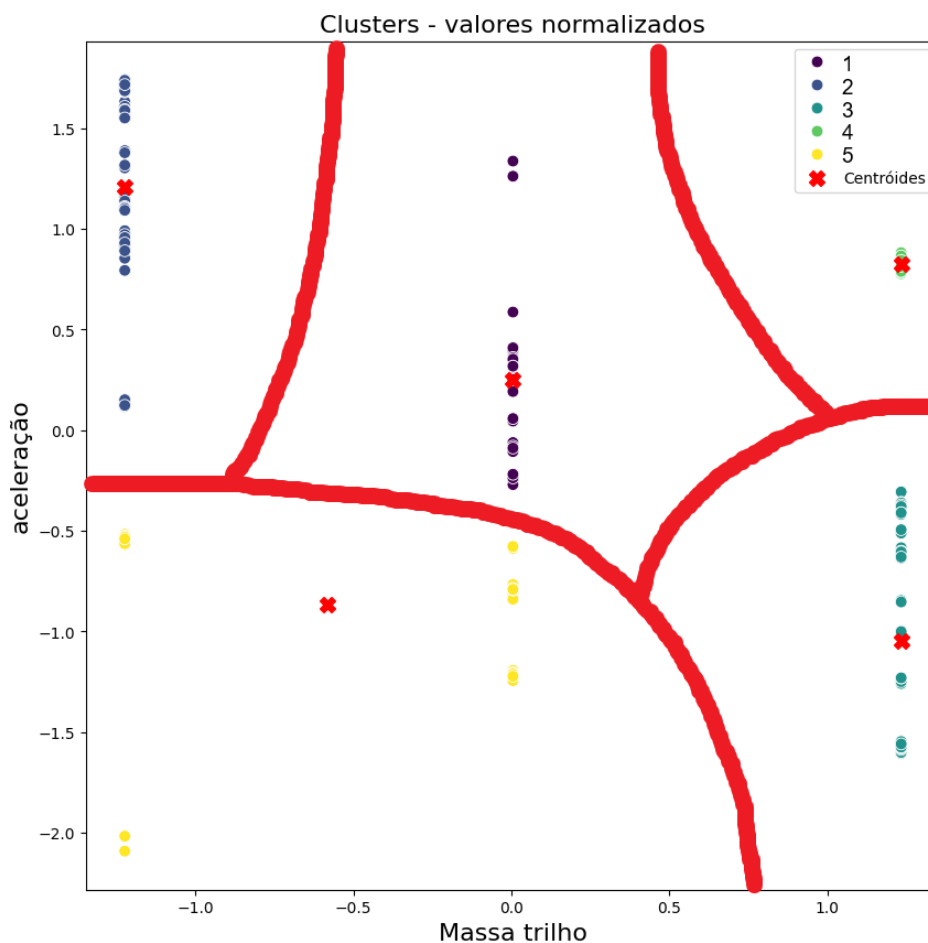
Na figura 10 é possível observar o gráfico que contém a forma como os dados foram agrupados através das fórmulas apresentadas nas seções 2, 2.1. Os clusters 1, 2, 3, 4 e 5 apresentados na tabela 2 estão associados respectivamente aos clusters 1, 2, 3, 4 e 5 da figura 10. Na figura 11 é possível analisar aproximadamente a rede poligonal de Thiessen e onde estão localizados seus centroides.

Figura 10- Análise de clusters obtidos



Fonte: Gabriela Molina Ciocci

Figura 11- Aproximação da rede poligonal de Thiessen do experimento



Fonte: Gabriela Molina Ciocci

Analizando os gráficos e a tabela 2, pode-se perceber que o cluster 4 representado pela figura apresentou um bom agrupamento da massa do corpo suspenso m_2 em função da aceleração e da massa do corpo no trilho de ar m_1 . No entanto, os outros clusters apresentaram uma dispersão muito grande para os valores de m_2 , o que indica uma dificuldade para a identificação correta da massa. Portanto, será necessário a diminuição de variabilidade de massas m_2 para melhorar sua probabilidade de identificação.

Com o intuito de aprofundar a análise da variabilidade dos dados coletados, optou-se por uma abordagem estatística utilizando o gráfico de Box Plot. Essa representação gráfica permite visualizar de forma clara a distribuição dos dados, identificando valores atípicos e comparando diferentes grupos. A seguir, são apresentados os resultados obtidos.

4.4 ANÁLISE DE DADOS UTILIZANDO BOX PLOT

O Box Plot, também conhecido como diagrama de caixa, é uma ferramenta visual poderosa para explorar a distribuição de um conjunto de dados. Essa representação gráfica é composta por uma caixa, que delimita a amplitude interquartil (AIQ), e por barras que se estendem a partir da caixa, indicando o intervalo total dos dados, excluindo os outliers. A linha horizontal dentro da caixa representa a mediana, que divide os dados em duas partes iguais. Os outliers, pontos que se encontram significativamente distantes dos demais dados, são representados por pontos ou asteriscos e podem influenciar a análise estatística. A posição da mediana em relação à caixa fornece informações sobre a simetria da distribuição: se a mediana estiver centralizada na caixa, a distribuição é considerada simétrica; caso contrário, ela é assimétrica. (SEABORN, 2012).

A figura 12 apresenta o Box Plot dos dados obtidos e nos mostra a distribuição dos valores, a amplitude e a simetria conforme a variação de m_2 .

Figura 12- Box Plot da aceleração em função de massa m_2 .



Fonte: Gabriela Molina Ciocci

Analisando a figura 12, pode-se perceber que entre os valores de m_2 entre a variação de 0,010kg até 0,022kg houve grande dispersão de aceleração. Essa variabilidade trouxe dificuldade para a identificação correta dos clusters e consequentemente, a provável diminuição da acurácia da Rede Neural.

Uma possível solução para o problema no momento é ajustar o número de valores da variável m_2 . Ao reduzir essa quantidade, os intervalos entre os valores se tornam maiores, o que pode facilitar a interpretação dos dados.

5 CONCLUSÃO PARCIAL E PRÓXIMAS ETAPAS

A coleta de dados experimentais, embora desafiadora, é crucial para obter um banco de dados robusto para realizar o treinamento eficaz da Rede Neural. Para aprimorar o modelo, o próximo passo será otimizar os valores de massa m_2 e adquirir novos dados com o trilho de ar inclinado, com diferentes angulações.

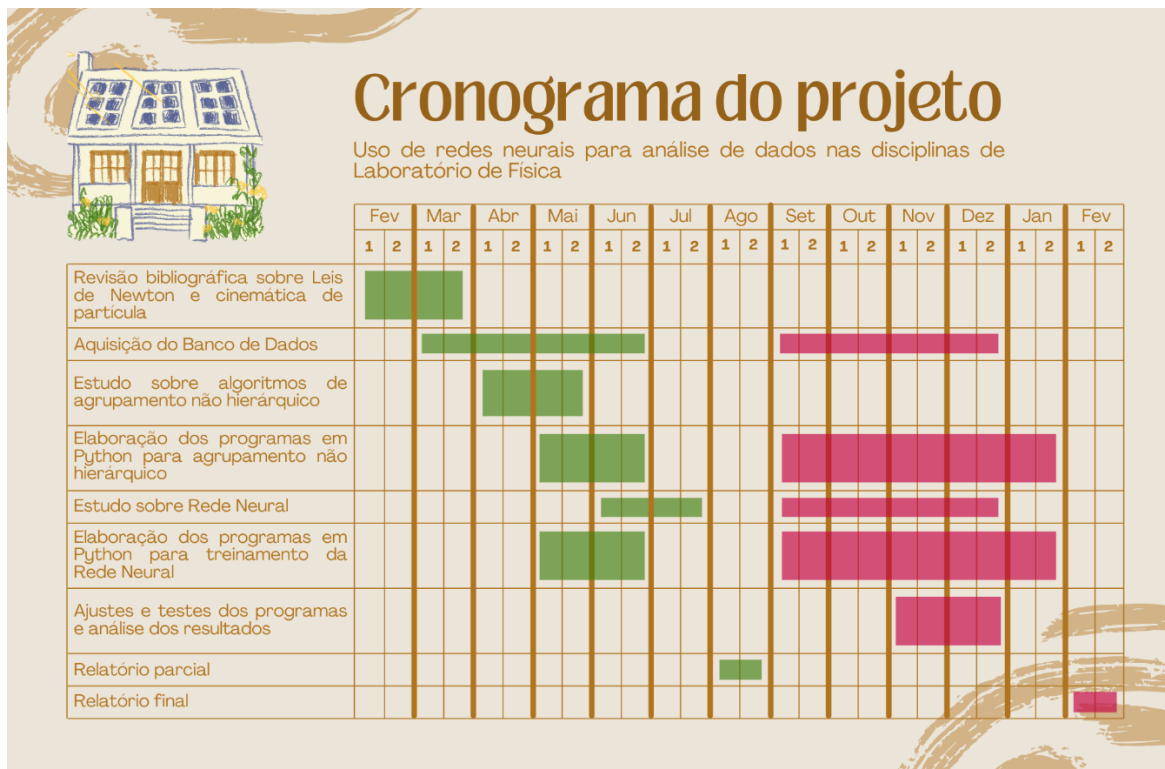
Assim, a expectativa é aumentar a quantidade dos dados, melhorar sua qualidade e obter uma melhor precisão da classificação, permitindo que a Rede Neural identifique corretamente a massa m_2 a partir dos dados de m_1 , ângulo e aceleração fornecidos, auxiliando nas aulas de laboratório de Física 1.

Até a conclusão do projeto, estima-se obter uma Rede Neural treinada para que em uma aula do Laboratório de Física 1 os alunos entrem com os valores obtidos e a Rede Neural retorne o valor da massa m_2 com precisão.

6 CRONOGRAMA

A figura 13 apresenta o cronograma das próximas etapas do projeto.

Figura 13 - Cronograma anual do projeto



Fonte: Gabriela Molina Ciocci

Tópicos que serão realizados nos próximos meses de projeto:

- Aquisição do Banco de Dados;
- Elaboração dos programas em Python para o treinamento e teste da Rede Neural;
- Estudos mais aprofundados sobre Rede Neural;
- Ajustes e testes dos programas;
- Análise de resultados obtidos;
- Relatório final.

REFERÊNCIAS

BARDINE, RENAN. (2000-2024). **NEURÔNIOS**. Disponível em: [Neurônios: Características, Funções, Estruturas e Tipos - Cola da Web](#). Acesso em: 03/09/2024.

BURROUGH PA, MCDONNELL R, LLOYD C. **PRINCIPLES OF GEOGRAPHICAL INFORMATION SYSTEMS**. 3RD ED. OXFORD UNIVERSITY PRESS; 2015, p. 160 – 161. Acesso em: 23/08/2024.

FÁVERO. **MANUAL DE ANÁLISE DE DADOS ESTATÍSTICA E MODELAGEM MULTIVARIADA COM EXCEL, SPSS E STATA**, v.9.1 – v.9.2, p. 300 – 327, 2022. Acesso em: 08/06/2024.

FETZER, J.H. (1990). **WHAT IS ARTIFICIAL INTELLIGENCE?.IN: ARTIFICIAL INTELLIGENCE: ITS SCOPE AND LIMITS. STUDIES IN COGNITIVE SYSTEMS, VOL 4**. SPRINGER, DORDRECHT. Disponível em: https://doi.org/10.1007/978-94-009-1900-6_1 . Acesso em: 22/08/2024.

GIORDANI, P., FERRARO, M.B., MARTELLA, F. (2020). **NON-HIERARCHICAL CLUSTERING. IN: AN INTRODUCTION TO CLUSTERING WITH R. BEHAVIORMETRICS: QUANTITATIVE APPROACHES TO HUMAN BEHAVIOR, VOL 1**. Disponível em: https://doi.org/10.1007/978-981-13-0553-5_3 . Acesso em: 11/08/2024.

HALLIDAY, David; RESNICK, Robert; WALKER, Jearl. **FUNDAMENTOS DE FÍSICA, VOLUME 1: MECÂNICA**. 10. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos Editora LTDA, 2016. v. 1. Acesso em: 25/05/2024.

HAMERLY, GREG; ELKAN, CHARLES 2002. ["ALTERNATIVES TO THE K-MEANS ALGORITHM THAT FIND BETTER CLUSTERINGS"](#) (PDF). PROCEEDINGS OF THE ELEVENTH INTERNATIONAL CONFERENCE ON

INFORMATION AND KNOWLEDGE MANAGEMENT (CIKM). Acesso em: 23/08/2024.

KUKREJA, H., BHARATH, N., SIDDESH, C. S., & KULDEEP, S. (2016). **AN INTRODUCTION TO ARTIFICIAL NEURAL NETWORK**. INT J ADV RES INNOV IDEAS EDUC, 1(5), 27-30. Disponível em: [AN-INTRODUCTION-TO-ARTIFICIAL-NEURAL-NETWORK.pdf \(researchgate.net\)](#). Acesso em: 03/09/2024.

LLOYD, STUART P. (1957). **"LEAST SQUARE QUANTIZATION IN PCM"**. BELL TELEPHONE LABORATORIES PAPER. PUBLISHED IN JOURNAL MUCH LATER: LLOYD, STUART P. (1982). Disponível em: ["Least squares quantization in PCM" \(PDF\). IEEE Transactions on Information Theory. 28 \(2\): 129–137. doi:10.1109/TIT.1982.1056489.](#) Acesso em: 03/09/2024.

MCCULLOCH, W.S., PITTS, W. (1943). **A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY**. *BULLETIN OF MATHEMATICAL BIOPHYSICS* 5, 115–133 Disponível em: <https://doi.org/10.1007/BF02478259>. Acesso em: 03/09/2024.

MOREIRA, C., (2013). **NEURÔNIO**. REV. CIÊNCIA ELEM., V1(1):008. Disponível em: <http://doi.org/10.24927/rce2013.008>. Acesso em: 03/09/2024.

MATPLOTLIB. **MATPLOTLIB.PYPLOTT**. 1991. Disponível em: https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html. Acesso em: 07/06/2024.

MEI/CNI, Abenge. **DIRETRIZES PARA O CURSO DE ENGENHARIA**. 2018. Disponível em: http://www.abenge.org.br/documentos/propostadcnabengemei_cni.pdf. Acesso em: 07/03/2024.

MITCHELL, Tom M. **MACHINE LEARNING**. v.3 - Decision Tree Learning. 1997. Disponível em: <https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf>. Acesso em: 01/02/2024.

PANDAS. **PANDAS DOCUMENTATION**. 2008. Disponível em: <https://pandas.pydata.org/docs/>. Acesso em: 07/06/2024.

PYTHON. **PYTHON 3.12.4 DOCUMENTATION**. 2001. Disponível em: <https://docs.python.org/3/>. Acesso em: 14/05/2024.

ROSENBLATT. FRANK. **THE PERCEPTRON, A PERCEIVING AND RECOGNIZING AUTOMATON PROJECT PARA. CORNELL AERONAUTICAL LABORATORY**. Disponível em: [rosenblatt-1957.pdf \(bpb-us-e2.wpmucdn.com\)](https://www.bpb-us-e2.wpmucdn.com/rosenblatt-1957.pdf). Acesso em: 03/09/2024.

SEABORN. **SEABORN DOCUMENTATION**. 2012. Disponível em: <https://seaborn.pydata.org/tutorial.html/> . Acesso em: 05/08/2024.

SCIKIT-LEARN. **SCIKIT-LEARN.ORG**. 2007. Disponível em: <https://scikit-learn.org/stable/about.html#history>. Acesso em: 07/06/2024.

SCIPY, api. **SCIPY DOCUMENTATION**. 2008. Disponível em: <https://docs.scipy.org/doc/scipy/>. Acesso em: 07/06/2024.

W. S. MCCULLOCH AND W. PITTS. (1943) **A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY**. THE BULLETIN OF MATHEMATICAL BIOPHYSICS, 5(4):115–133. Acesso em: 03/09/2024.

YELLOWBRICK. **ELBOW METHOD**. 2016. Disponível em: <https://www.scikit-yb.org/en/latest/api/cluster/elbow.html>. Acesso em: 08/06/2024.

ZHANG, Z. **ARTIFICIAL NEURAL NETWORK. IN: MULTIVARIATE TIME SERIES ANALYSIS IN CLIMATE AND ENVIRONMENTAL RESEARCH**. 2018.

Disponível em: https://doi.org/10.1007/978-3-319-67340-0_1. Acesso em: 11/08/2024.

ZHOU, ZHI-HUA. **MACHINE LEARNING**. SPRINGER. 2021. Disponível em: <https://doi.org/10.1007/978-981-15-1967-3>. Acesso em: 22/08/2024.