

Projeto de Graduação



1 de Julho de 2024

TÉCNICAS DE MACHINE LEARNING E CONTROLE EXTREMAL APLICADOS NA DETECÇÃO DE EXOPLANETAS

Bruna Vianna de França Costa



www.ele.puc-rio.br

Projeto de Graduação



TÉCNICAS DE MACHINE LEARNING E CONTROLE EXTREMAL APLICADOS NA DETECÇÃO DE EXOPLANETAS

**Aluno: Bruna Vianna de França
Costa**

**Orientador: Eduardo Costa da Silva
Co-Orientador: William de Souza Barbosa**

Trabalho apresentado como requisito parcial para conclusão do curso de Engenharia de Controle e Automação da Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.

Agradecimentos

O curso de Engenharia de Controle e Automação foi uma trajetória emocionante e desafiadora. Dito isto, gostaria de agradecer a todos os envolvidos. Não apenas a professores, mas todos dentro do campus que fizeram da universidade a minha casa.

Aos professores da PUC-Rio que me fizeram amadurecer e construir a pessoa que sou hoje, cheia de sonhos e energia. Obrigada por responderem a curiosidade de uma adolescente de ensino médio que buscava conhecer e entender o mundo.

Ao meu orientador, Prof. William Barbosa que me apoiou, ensinou e permitiu que eu demonstrasse minhas cores durante o curso de Engenharia de Controle e Automação na PUC-Rio.

Graças a todos vocês, mesmo com falhas e erros, me transformaram em quem eu sou, formando as estrelas mais brilhantes da constelação da minha vida para uma nova jornada.

Queria agradecer a minha família, em específico, a minha mãe por me proporcionar uma educação de qualidade e acreditar em mim e nos meus sonhos desde criança. Não há palavras que descrevam a honra que tenho de ter tido todo o apoio durante anos e a honra de ser filha de uma mulher brilhante.

A amigos que me apoiaram durante toda a trajetória, me ouviram reclamar, chorar e brilhar. Obrigada por estarem comigo nessa trajetória.

A música e arte que me fizeram não me perder durante a jornada, me dando forças, esperanças e inspirações. Graça a vocês estou aqui, viva e esperançosa para o futuro.

E por fim, a mim, pela força e determinação, olhando para as dificuldades com sarcasmo e um sorriso. Lembre-se, a única vez que devo olhar para trás é para ver o quão longe cheguei.

Como Audrey Hepburn diria: E antes que eu me vá, com gratidão e amor, eu termino. PUC, você realmente foi um grande amigo.

Resumo

A NASA™ desenvolveu o Telescópio Kepler usando observações de eventos de trânsito, que são pequenas quedas no brilho de uma estrela quando um planeta passa na frente dela.

Como os eventos de trânsito duram apenas uma fração do dia, todas as estrelas devem ser monitoradas continuamente, implicando que seus brilhos devem ser medidos pelo menos uma vez a cada poucas horas. A capacidade de visualizar continuamente as estrelas monitoradas determina que o campo de visão nunca deve ser bloqueado em qualquer época do ano. Tais condições, principalmente no espaço, são difíceis de prever, por isso, alguns casos de falso positivos e negativos ocorrem na hora da classificação de exoplanetas.

A fim de aperfeiçoar essa classificação, tratando o problema de falsos positivos e negativos, foi desenvolvido dois tipos de abordagens de modelos preditivos para as observações de exoplanetas feita pelo telescópio Kepler.

Para tal desenvolvimento foi utilizado um sistema *Model Free*, baseado em modelos de aprendizagem de máquina usando Redes Neurais e Rede Neuro-Fuzzy, e também modelos mais matemáticos, como Controle Extremal.

Para o modelo de Redes Neurais, foram escolhidos e comparados dois modelos feitos em Python™: Perceptron e Multi-Layer Perceptron. Desses modelos, foi utilizado o com melhor resultado envolvendo métrica de acurácia, precisão, recall e RMSE.

Para o modelo de Redes Neuro-Fuzzy, foram escolhidos e comparados dois modelos distintos usando duas bases de dados diferentes desenvolvido também em MATLAB™. Desses modelos, foi utilizado o com melhor resultado envolvendo métrica de acurácia e RMSE.

Para o modelo de Controle Extremal foi modelado um sistema em MATLAB™ com mesmo ruído que o a base de dados, escolhida uma função de custo, e em seguida foi aplicado uma entrada senoidal e construído um controle eficiente para analisar a melhor resposta do sistema.

Palavras-chave: **Aprendizado de Máquina, ANFIS, Perceptron, Perceptron de Múltiplas Camadas, Controle Extremal, Base de Dados, Telescópio Kepler.**

MACHINE LEARNING AND EXTREMUM SEEKING CONTROL TECHNIQUES APPLIED TO THE DETECTION OF EXOPLANETS

Abstract

NASA™ developed the Kepler Telescope using observations of transit events, which are small dips in a star's brightness when a planet passes in front of it.

Since transit events last only a fraction of the day, all stars must be monitored continuously, implying that their brightnesses must be measured at least once every few hours. The ability to continuously view monitored stars means that the field of view must never be blocked at any time of the year. Such conditions, especially in space, are difficult to predict, therefore, some cases of false positives and negatives occur when classifying exoplanets.

In order to improve this classification, dealing with the problem of false positives and negatives, it was developed two types of predictive model approaches for exoplanet observations made by the Kepler telescope.

For this, a Model Free system was used, based on machine learning models using Neural Networks and Neuro-Fuzzy Network, and also more mathematical models, such as Extremum Seeking Control.

For the Neural Networks model, two of them were made, chosen and compared in Python™: Perceptron and Multi-Layer Perceptron. Of these models, the one with the best results was chosen based on best performing accuracy, precision, recall and RMSE metrics.

For the Neuro-Fuzzy Networks model, it was developed in MATLAB™ two models with completely different databases. Of these models, the chosen one was with the best result involving accuracy metrics and RMSE.

For the Extremum Seeking Control model, a system was made in MATLAB™ with the same noise as the data. This noise was then applied as a cost function. After that, a sinusoidal input was applied and with that, it was constructed an efficient control system to analyze the best response.

Keywords: Machine Learning, ANFIS, Perceptron, Multi-Layer Perceptron, Extremum Seeking Control, Database, Kepler Telescope.

Sumário

1	INTRODUÇÃO	1
2	ESTADO DA ARTE - Parte 1: "O ESPECTADOR"	7
3	ESTADO DA ARTE: "A CURADORIA"	8
a	Exoplanetas	8
b	Telescópio	8
c	Curvas de Luz	10
d	Ruídos Instrumentais	11
1	Crosstalk	11
2	Bandas Rolantes	11
3	Raios Cósmicos e Queda Repentina de Sensibilidade de Pixel	11
4	Downlink em Dados	11
5	Argabrightening	11
6	Reação das rodas	12
e	Ruídos de Imagem	12
4	ESTADO DA ARTE: "O ARTISTA"	13
a	Sistemas baseado em Modelos	13
b	Sistemas sem Modelo	13
c	Sistemas Inteligentes	13
1	Rede Neural	15
d	Tipos de Redes Neurais	16
1	Single Layer Perceptron	16
2	MultiLayer Perceptron	17
3	Single Layer Perceptron vs. MultiLayer Perceptron	18
4	Lógica Fuzzy	19
5	Rede Neuro-Fuzzy	20
6	ANFIS	21
e	Métricas de avaliação de desempenho	21
f	Hiperparâmetros de Redes Neurais	24
g	Visão Computacional	27
h	Teoria de Controle Clássica	28
1	Planta	28
2	Tipo de Malha do Sistema	28
3	Controle PID	29
i	Controle Extremal	29
j	Função de Custo	32
1	ISE	32
2	IAE	33
3	ITAE	33
k	Ferramentas de Avaliação de Sistemas	33
1	Retrato de Fase	34
2	Diagrama de Bode	34
5	DISCUSSÃO: "A ARTE"	36
a	Base da Dados: Rede Neural	36
b	Rede Neural	36
c	Base da Dados: Rede Neuro-Fuzzy	37
d	Rede Neuro-Fuzzy	38
e	Base de dados Controle Extremal	39
f	Controle Extremal	40
6	RESULTADOS: "A EXPOSIÇÃO"	42
a	Rede Neural	42
b	Rede Neuro-Fuzzy	43
c	Controle Extremal	45
1	Desempenho para a Função de Custo ISE	46
2	Desempenho para a Função de Custo IAE	49
3	Phase Portrait Plotter	51

Lista de Figuras

1	O Pequeno Príncipe™ [1].	1
2	O baobá [2].	2
3	O astrônomo de O Pequeno Principe e o Telescópio [3].	2
4	Dentro da Instalação de Processamento de Riscos da Astrotech em Titusville, Flórida, antes de seu lançamento, a espaçonave Kepler da NASA™é colocada em um suporte para abastecimento [4].	3
5	Planetas descobertos na varredura da missão Kepler da NASA™em busca de planetas pequenos e habitáveis nos últimos nove anos [5].	4
6	O Telescópio Espacial Kepler da NASA™orbita o Sol em conjunto com a Terra, afastando-se lentamente da Terra.	4
7	Curva de luz de um planeta em trânsito em sua estrela.	5
8	Kepler Fotometro [6].	9
9	Campo com o Kepler FOV sobreposto [6].	9
10	Exemplo de curva de luz extraído durante o desenvolvimento do atual estudo.	10
11	Representação do HAL 9000 e a Siri.	14
12	Representação visual de um neurônio humano. [7]	15
13	Exemplo de Rede Neural com três camadas internas.	16
14	Representação visual de um Perceptron.	17
15	Representação visual de um MultiLayer Perceptron com três camadas.	18
16	Exemplo de Funções de Pertinência [8].	20
17	Representação de uma matriz de cofusão gerada no atual estudo. Nesta matriz foi utilizada um MLP de duas camadas e função de ativação RELU.	23
18	Visualização de funcionamento da taxa de aprendizado alto e baixo.	24
19	Visualização da função Softmax.	25
20	Visualização da função Log-Softmax.	25
21	Visualização da função Tanh.	26
22	Visualização da função Hardtanh.	26
23	Visualização da função ELU.	27
24	Diagrama de blocos de um sistema de Controle Genérico.	28
25	Exemplo de Malha Aberta de um sistema.	28
26	Exemplo de Malha Fechada de um sistema.	29
27	Diagrama de blocos de um sistema final de controle extremal.	30
28	Curva genérica de consumo de combustível ao longo do tempo.	31
29	Diagrama em blocos do controle extremal [9]	32
30	Exemplo de Diagrama de Bode. [10]	35
31	Formato da análise de parâmetros do BLS	37
32	Periodograma do K2-3 [11].	39
33	Arquivo em pixel da amostra da EPIC 200084891 extrai do LightKurve™.	40
34	Arquivo em pixel da amostra da EPIC 211741417 extrai do LightKurve™.	40
35	Função de perda durante treinamento.	43
36	Função de perda durante validação.	43
37	Simulação do erro da base Kaggle™	45
38	Simulação do erro base Astropy™	45
39	Diagrama em blocos do sistema de controle extremal.	46
40	Função de Custo ISE com baixa taxa de aprendizado, $r\tau = 4$	47
41	Função de Custo ISE com frequência de 600rad/s, $r\tau = 4$	47
42	Função de Custo ISE com demodulação de 0.5, $r\tau = 4$	48
43	Simulação de Controle Extremal $r\tau = 4$	48
44	Simulação de Controle Extremal $r\tau = 5$	49
45	Função de Custo IAE com taxa de aprendizado de 0.4, $r\tau = 4$	49
46	Função de Custo IAE com taxa de aprendizado de 0.4, $r\tau = 4$	50
47	Função de Custo IAE com filtro passa-alta de frequência 1 rad/s, $r\tau = 4$	50
48	Simulação de Controle Extremal $r\tau = 4$	51
49	Simulação de Controle Extremal $r\tau = 4$	51
50	Phase Portrait Plotter $r\tau = 4$	52
51	Phase Portrait Plotter $r\tau = 5$	52

Lista de Tabelas

1	Lista de tipos de rede e resultados	42
2	Resultados das Matrizes de Confusão	42
3	Avaliação da rede ANFIS com a base Kaggle™ por meio do RMSE	44
4	Avaliação da rede ANFIS com a base Kaggle™ por meio do MSE	44
5	Avaliação da rede ANFIS com a base Kaggle™ por meio da Acurácia	44
6	Parâmetros do sistema de controle extremal	48
7	Parâmetros do sistema de controle extremal	51

Lista de Abreviações

ANFIS - Adaptive Network-based Fuzzy Inference System
BLS - Box Mínimos Quadrados
CCD - Charge-Coupled Device
ESC - Extremum Seeking Control
FN - Falso Negativo
FOV - Field of View
FP - Falso Positivo
JARVIS - Just A Rather Very Intelligent System
KOI - Kepler Object of Interest
LC - Long-Cadence
MLP - MultiLayer Perceptron
MSE - Erro Quadrático Médio
NASA - National Aeronautics and Space Administration
ONU - Organização das Nações Unidas
ReLU - Rectified Linear Unit
RMSE - Root Mean-Square Error
SC - Short-Cadence
SDA - Science Data Acumulador
SGD - Gradiente Descendente Estocástico
SLP - Single Layer Perceptron
TESS - Transiting Exoplanet Survey Satellite
UA - Unidade Astronômica
VN - Verdadeiro Negativo
VP - Verdadeiro Positivo

1 INTRODUÇÃO

A história de "O Pequeno Príncipe"™ de Antoine de Saint-Exupéry gira em torno de um avião que se vê perdido no deserto do Saara. Lá, ele encontra o Pequeno Príncipe. Nesse encontro, o personagem principal, relata ter vindo de um pequeno asteroide chamado B-612, onde ele viveu sozinho com uma rosa. A rosa era sua amiga e companheira, mas também muito vaidosa e exigente. Desiludido com a atitude da rosa, o pequeno príncipe parte em uma jornada por diferentes asteroides até chegar na parada final, a Lua. [12]

A curiosidade do ser humano não difere muito do Pequeno Príncipe™. A prova disso está em um dos eventos históricos mais impactantes da Guerra Fria, a Corrida Espacial. Com intuito de promover a exploração do espaço, Estados Unidos e a União Soviética, incetivaram o desenvolvimento científico, realizando expedições tripuladas ao espaço que no final levou o homem a lua, transformando assim literatura em realidade.



Figura 1: O Pequeno Príncipe™ [1].

Ao chegar a lua, o pequeno príncipe conhece um geógrafo que lhe explica que a Lua é uma esfera seca e sem vida, e que não há ninguém para explorá-la ou cuidar dela. O pequeno príncipe fica um pouco decepcionado, pois esperava encontrar algo mais interessante. Como o personagem principal, o marco da Lua para civilização atual foi o primeiro passo, a primeira vitória do ser humano na área de Exploração Espacial, porém não é a última. O espaço é desconhecido e com os problemas ambientais atuais, a exploração espacial pode fornecer ideias cruciais sobre como os seres humanos podem sobreviver diante as mudanças catastróficas no clima. [13]

No livro há a presença de uma metáfora importante para o desenvolvimento do presente estudo, os baobás. Eles são árvores gigantes e muito fortes, que podem crescer em diversos planetas. No entanto, o pequeno príncipe alerta sobre a necessidade de cuidar para que essas árvores não cresçam descontroladamente. Ele explica que se os baobás não forem arrancados enquanto ainda são pequenos, eles se tornarão enormes e suas raízes poderão destruir o asteroide.



Figura 2: O baobá [2].

Como os baobás, se o aquecimento global não for tratado o quanto antes, as mudanças climáticas serão irreversíveis. De acordo com o secretário-geral da ONU™, António Guterres: *"Ainda é possível limitar o aumento da temperatura global a 1,5°C e evitar o pior da mudança climática. Mas somente com ações climáticas drásticas e imediatas."* [14]. Ainda é possível salvar a Terra, contudo o que mais falam é procurar outro planeta habitável. Marte está em primeiro lugar, porém não é a única solução.

No conto, o pequeno príncipe usa um telescópio para observar os asteroides em seu pequeno asteróide. Ele o utiliza para olhar para o céu e observar os diferentes corpos celestes que passam por perto. Através do telescópio, o Pequeno Príncipe se conecta com o vasto universo ao seu redor, alimentando sua curiosidade e senso de maravilha.



Figura 3: O astrônomo de O Pequeno Príncipe e o Telescópio [3].

Paralelamente, em 2009 a NASA™lançou o projeto espacial chamado Telescópio Kepler, nomeado em homenagem o cientista Johannes Kepler. Esse projeto foi designado para pesquisar a região da Via Láctea fora do sistema solar em busca de exoplanetas. [15]

Exoplanetas ou planetas extrassolares, são planetas que orbitam estrelas fora do sistema solar em que a Terra está inserida. Eles podem variar em tamanho, composição e características físicas. Alguns são rochosos, semelhantes à Terra, enquanto outros são gigantes gasosos, comparáveis a Júpiter e Saturno.

Além disso, muitos deles orbitam estrelas em zonas habitáveis, onde as condições poderiam permitir a existência de água líquida e, portanto, potencialmente suportar a vida humana [16].

O telescópio foi montado para procurar exoplanetas com metade ou até duas vezes o tamanho da Terra na zona habitável de suas estrelas, onde a água líquida poderia existir na superfície do planeta. Dentro dos alvos analisados existem observações que encontram-se nesses parâmetros, como o caso do Kepler-186f, indicando assim a existência de outros locais promissores para a vida humana futuramente.



Figura 4: Dentro da Instalação de Processamento de Riscos da Astrotech em Titusville, Flórida, antes de seu lançamento, a espaçonave Kepler da NASA™ é colocada em um suporte para abastecimento [4].

O telescópio Kepler encontra-se na sua segunda missão desde 2014 e já rotulou mais de 10.000 possíveis exoplanetas. Em 2017, foram confirmados mais de 3.000 exoplanetas no total. Em 2018, depois de nove anos de atividade, ele ficou sem combustível necessário para futuras operações científicas. Visando esses problemas, Kepler foi sucedido pelo *Transiting Exoplanet Survey Satellite* (TESS), que continua a busca por exoplanetas [15].

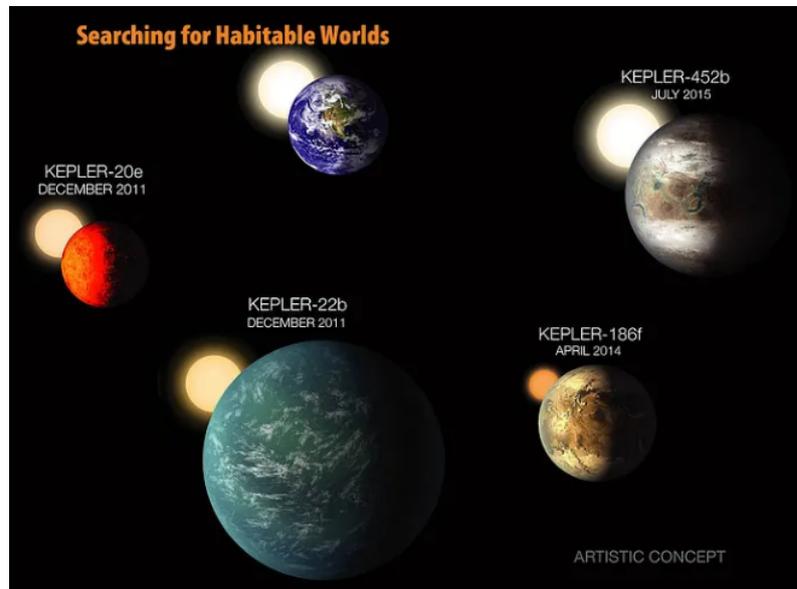


Figura 5: Planetas descobertos na varredura da missão Kepler da NASA™ em busca de planetas pequenos e habitáveis nos últimos nove anos [5].

A Figura 5 mostra os planetas que o telescópio Kepler encontrou. O primeiro é o Kepler-20e que foi descoberto em dezembro de 2011 orbitando uma estrela semelhante ao Sol, ligeiramente mais fria e menor. O Kepler-22b foi anunciado no mesmo mês, sendo um planeta na zona habitável de uma estrela semelhante ao Sol, mas tem mais do dobro do tamanho da Terra. Como citado acima, o Kepler-186f foi descoberto em 2014 e é o primeiro planeta do tamanho da Terra encontrado na zona habitável de uma pequena e fria estrela com cerca de metade do tamanho e massa do Sol. Por último, o Kepler-452b é o primeiro planeta próximo do tamanho da Terra na zona habitável de uma estrela muito semelhante ao Sol [5].

O Telescópio Espacial Kepler foi colocado em uma órbita heliocêntrica ao redor do Sol [17], com um período de aproximadamente 372.5 dias. Nesta órbita a espaçonave se afasta lentamente da Terra e até últimas atualizações, estava a uma distância de mais de 0,4 UA.

Devido a órbita, Kepler aponta continuamente para uma única região do céu, isso permite maior estabilidade evitando as altas dosagens de radiação associadas à órbita da Terra. [6]

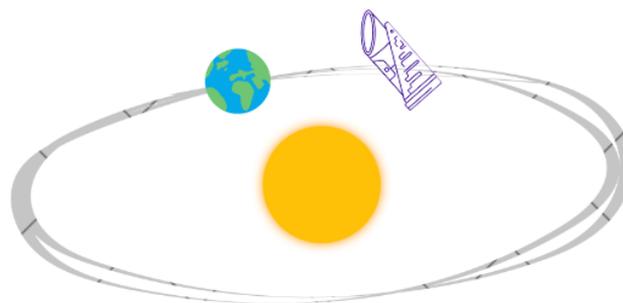


Figura 6: O Telescópio Espacial Kepler da NASA™ orbita o Sol em conjunto com a Terra, afastando-se lentamente da Terra.

O "coração" do telescópio Kepler reside em seu sofisticado fotômetro, um instrumento sensível concebido para capturar uma gama específica de luz conhecida como espectro de luz visível. Ao contrário dos telescópios tradicionais que se concentram num único alvo, o Kepler adotou uma abordagem de campo amplo, monitorizando o brilho de mais de 100.000 estrelas numa determinada área do céu.

Os planetas, por fim, foram detectados observando pequenas quedas no brilho de uma estrela. Para essa queda acontecer, o Kepler olhava para uma área específica do céu, monitorando continuamente o brilho dentro de seu campo de visão até que um planeta passe na frente dela. Esse fenômeno é conhecido como evento de trânsito. Durante um trânsito, o planeta bloqueia temporariamente uma pequena fração da luz da estrela [16] [18].

Os instrumentos do Kepler mediram o brilho das estrelas ao longo do tempo. Quando ocorreu um evento de trânsito, houve uma pequena queda no brilho da estrela. Isto foi registrado no que é chamado de "curva de luz", que é essencialmente um gráfico do brilho de uma estrela ao longo do tempo, como indicado abaixo na Figura 7.

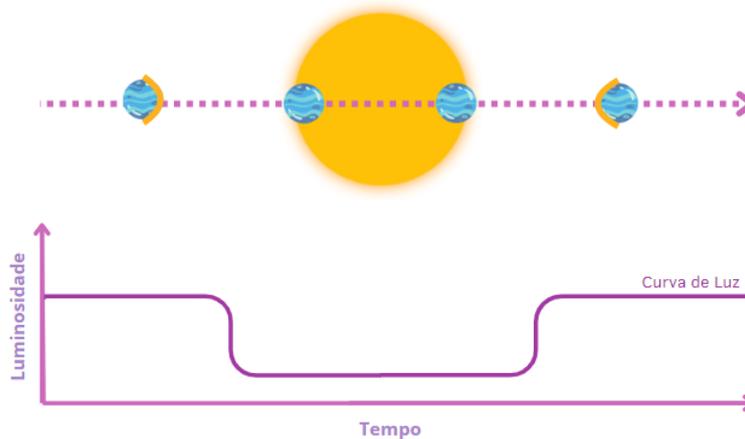


Figura 7: Curva de luz de um planeta em trânsito em sua estrela.

Planetas em órbitas próximas em torno de suas estrelas produziram quedas regulares e periódicas no brilho à medida que passassem na frente da estrela. Esses padrões repetidos eram fortes indicadores de um exoplaneta em potencial.

Cientistas, por fim, analisaram a enorme quantidade de dados coletados pelo Kepler. Eles procuraram padrões repetidos nas curvas de luz e usaram algoritmos para identificar potenciais candidatos a exoplanetas. Depois que o telescópio identificou um potencial candidato, observações de acompanhamento foram frequentemente feitas usando outros telescópios para confirmar as descobertas.

Outro detalhe interessante, com as leis do movimento planetário do cientista Johannes Kepler, os cientistas foram capazes de determinar o tamanho e o período orbital dos exoplanetas. Outros dados, tais como observações espectroscópicas, poderão fornecer informações adicionais sobre a atmosfera e composição dos exoplanetas.

Uma vez confirmado um candidato a exoplaneta, este era adicionado à lista de observações confirmadas e as suas características eram catalogadas. E foi assim que o Kepler descobriu e catalogou milhares de exoplanetas.

As observações do Kepler desafiaram os modelos estabelecidos de formação e migração planetária, levando a uma reavaliação da nossa compreensão sobre as possibilidades que apenas os filmes e livros sobre o espaço sonhavam.

Como indicado acima, o método de trânsito utilizado é especialmente eficaz para encontrar planetas que estão relativamente próximos das suas estrelas hospedeiras e têm períodos orbitais curtos. Contudo, é menos eficaz para planetas com longos períodos orbitais ou aqueles em órbitas altamente inclinadas, onde os trânsitos são menos prováveis de ocorrer a partir do ponto de vista do Kepler.

O catálogo de candidatos a planetas do Kepler também é conhecido como KOI. No entanto, como citado acima, os KOIs também incluem eventos classificados como falsos positivos astrofísicos, além de rúidos, gerando um grau de incerteza na avaliação final [19].

NASA™ e outros pesquisadores como Alka Kumari [20] e Margarita Bugueño et al. [21] já desenvolveram projetos, que usam técnicas de Aprendizado de Máquina, Aprendizado Profundo, Aprendizagem Supervisionada e até Engenharia de Recursos para ajudar na classificação das amostras do telescópio e eliminar o efeito dos falsos positivos e negativos. Todos os projetos, trazem uma acurácia próxima ou maior que 0.95, mostrando que o tratamento com tais ferramentas são uma boa solução. Ademais, projetos como o da Blenda Guedes [22] e de Jobanbir Singh et al [22] trazem outra abordagem usando Rede Neural e Rede Neural Profunda.

No atual estudo, pretende-se aprofundar esses projetos, usando novas abordagens com Rede Neuro-Fuzzy, *Model Free*, e usando modelos matemáticos a partir do Controle Extremal. Busca-se então comparar esses os dois modelos e ver qual apresenta melhores métricas de resultado na hora de classificar um exoplaneta. Também, nesse estudo deseja-se melhorar a acurácia de forma a não colocar hiperparâmetros que sobrecarreguem os modelos.

Como no livro *O Pequeno Príncipe™*, o atual estudo busca explorar não apenas exoplanetas mas a curiosidade e imaginação para solucionar os problemas expostos e eventualmente abrir portas para a possibilidade de colonização ou migração humana para outros sistemas estelares.

O presente estudo está dividido em quatro seções. Na primeira é apresentada a seção de Estado da Arte, que apresenta os tópicos necessários para a compreensão do trabalho, como o entendimento de diferentes modelos de Rede Neuro-Fuzzy e métricas de estatística, e também os conceitos da teoria de Controle Extremal necessários ao entendimento do problema. Na seção seguinte, a de Desenvolvimento, é analisada a performance da detecção de exoplanetas usando as ferramentas citadas acima. Em continuidade, são apresentados os Resultados obtidos e estudos comparativos do desempenho dos sistemas preditivos propostos. A última seção apresenta as conclusões e sugestões para trabalhos futuros.

2 ESTADO DA ARTE - Parte 1: "O ESPECTADOR"

Esse tópico foi criado a fim de entender melhor a percepção do leitor em relação ao tema apresentado nesse estudo. Durante o período de estruturação da ideia deste trabalho de fim de curso foi perceptível a reação das pessoas ao serem apresentadas ao tema, demonstrando dúvidas de conceitos que são base para o estudo. Com isso, foi notada a possibilidade de entender melhor o público e testar se efetivamente os conhecimentos foram repassados.

Em primeira instância, iria ser realizado um estudo que atuaria em duas partes. A primeira onde é realizada a pergunta, por exemplo, "Você sabe o que é um exoplaneta?". Caso o usuário não souber, ele vai para segunda parte, onde é encaminhado para uma aba onde contém o texto do atual estudo explicando o conceito. Assim que explicado, é perguntado novamente o entendimento. Se continuou não entendendo, pede-se para explicar em um breve texto o porque, caso contrário segue com mais perguntas.

Note que, o estudo teria uma abordagem mais introdutória, logo ele foi realizado com pessoas que não conhecem a área ou tem um pouco de conhecimento sobre. Foi desenvolvido um pequeno formulário usando o Google Forms™ para obter as respostas e abaixo estão os resultados encontrados:

Porém, devido ao prazo de entrega, não foi completado essa etapa. Foi feita uma análise rápida mostrando o estudo as pessoas. Foi exigido que elas lêssem e tirassem dúvidas. Por meio dessas dúvidas foi entendido os pontos de importância para melhora. Logo, o atual estudo já foi desenvolvido tentando atender a necessidade do público, para melhor entendimento.

3 ESTADO DA ARTE: "A CURADORIA"

a Exoplanetas

Como dito anteriormente, um exoplaneta é qualquer planeta além do nosso sistema solar. Boa parte orbita outras estrelas, mas os planetas rebeldes, não estão vinculados a qualquer uma estrela. Uma grande maioria dos exoplanetas são encontrados através de métodos indiretos como o método de trânsito ou monitorando o espectro de uma estrela em busca de sinais que forneçam informações de um planeta. Os telescópios espaciais encontraram milhares de planetas observando "trânsitos" [23].

A principal característica que se destaca em planetas extrasolares é a diversidade. Os exoplanetas preenchem muito mais o espaço de parâmetros, com órbitas que variam de 0.03 a 5.5 UA, com excentricidades entre 0 e 0.8, e massas entre $15M_{Terra}$ e $15M_{Jupiter}$. Dentre os dados que são disponibilizados pela NASA™, existem poucos casos com medidas por outros métodos, para os quais um conjunto maior de parâmetros tenha sido determinado, o que permitiria uma análise mais profunda [18].

De acordo com a Letícia Rioga do Núcleo de Astronomia da Universidade Federal de Minas Gerais: "Uma forma a ser citada mas que não entrará no estudo de detecção de exoplanetas é observando a variação na posição da estrela ao longo do tempo. Esse método, chamado de variação da velocidade radial, só possibilita a observação de planetas gasosos com tamanhos próximos ou maiores que o de Júpiter. Quanto maior o tamanho do planeta e maior a proximidade dele com a estrela, mais a estrela parecerá estar dançando, ao observarmos a variação de sua posição ao longo do tempo. Isso está relacionado ao centro de massa. O centro de massa é o ponto em que um objeto pode ser equilibrado. Por exemplo, se você tentar equilibrar um cabo de madeira em cima de seu dedo, estando o cabo na horizontal, você provavelmente posicionará o dedo no meio do cabo, onde está seu centro de massa" [24].

Atualmente com avanço das pesquisas e desenvolvimento de ferramentas inteligentes que facilitam a análise dessas informações, existem avanços significativos nas análises mais profundas das constituições desses planetas.

David S. Spiegel et.al, revisam a ampla gama de estruturas internas que são conhecidas em sistemas exoplanetários, desde observações que são mais de 10 vezes mais massivos que Júpiter, até objetos de massa intermediária semelhantes a Netuno, com núcleos grandes e envelopes moderados de hidrogênio/hélio até para objetos rochosos com aproximadamente a massa da Terra [25].

b Telescópio

O telescópio Kepler tem uma abertura de 0.95 metros e um campo de visão de 115.6 graus^2 . Ele é composto de único instrumento, um fotômetro, que possui um conjunto de 42 CCDs dispostos em 21 módulos. CCDs são circuitos integrados sensível à luz que capturam imagens convertendo fótons em elétrons. Metade da passagem de banda máxima é de 435 a 845 nm, com > 1% de resposta espectral relativa. Cada CCD de 50x25 mm possui 2200x1024 pixels [6]. O intervalo entre leituras de um determinado pixel de um CCD é composto por um tempo de exposição definido em 6.019803 segundos e um tempo de leitura fixo de 0.51895 segundos. Para cada integração, todos os pixels foram lidos e somados temporalmente no *Science Data Acumulador*(SDA).

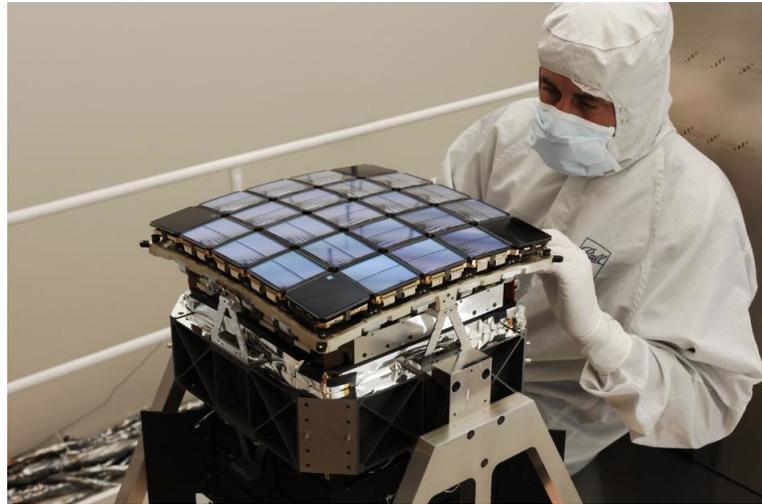


Figura 8: Kepler Fotometro [6].

Em média, 32 pixels são lidos do SDA por alvo. Os dados de cadência longa (LC) são somados em compartimentos de 29.424 minutos (270 integrações), enquanto os dados de cadência curta (SC) são somados em compartimentos de um minuto (9 integrações) [26].

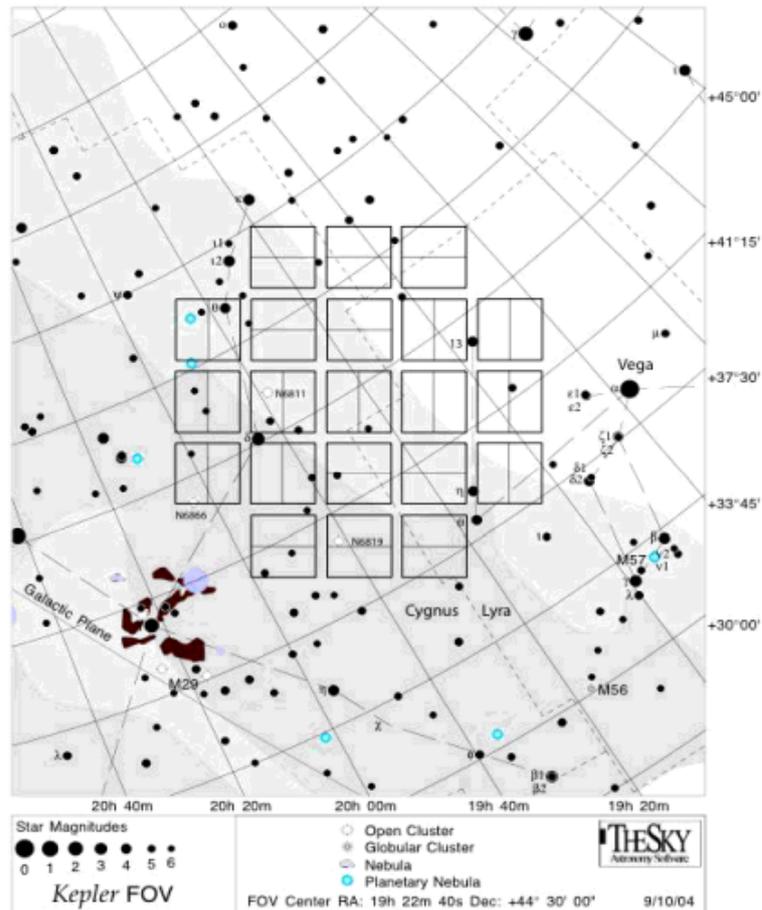


Figura 9: Campo com o Kepler FOV sobreposto [6].

A Figura 9 mostra o campo com o Kepler FOV sobreposto. Os quadrados mostram o FOV de 5 graus quadrados de cada um dos 21 módulos. Os CCDs cobrem um padrão simétrico de quatro direções no céu, de modo que a maioria das mesmas estrelas permanecem visíveis durante a missão, mesmo depois de uma rotação trimestral de 90°. Além disso, a orientação das linhas e colunas de cada localização do módulo no céu é preservada para todas as orientações de rotação, exceto para o módulo central, que é apenas 180 graus simétricos. O rolo é necessário para manter os painéis solares orientados para o Sol e o radiador apontava para o espaço profundo [26].

c Curvas de Luz

Arquivos de curva de luz são produzidos para cada alvo usando fotometria de abertura simples. Para qualquer trimestre específico, há mais de 160.000 alvos de cadência longa e até 512 alvos de cadência curta sendo arquivados. Os alvos de cadência curta têm sempre uma curva de luz de cadência longa correspondente. Alvos de cadência longa serão observados por pelo menos um quarto e as metas de cadência curta serão observadas por pelo menos um mês (exceto para o quarto trimestre, onde os alvos no módulo 3 foram perdidos devido a uma falha de hardware). No caso em que um alvo é observada em cadência longa e curta, haverá uma curva de luz de cadência longa a cada trimestre e acima a três curvas de luz de cadência curta, uma para cada mês. Um arquivo de curva de luz contém dados de séries temporais. Cada ponto de dados corresponde a uma medição de um longo ou cadência curta. Os dados de cadência longa e curta não são misturados em um determinado arquivo de curva de luz. Para cada ponto de dados existem vários valores de fluxo e centróide junto com incertezas [26].

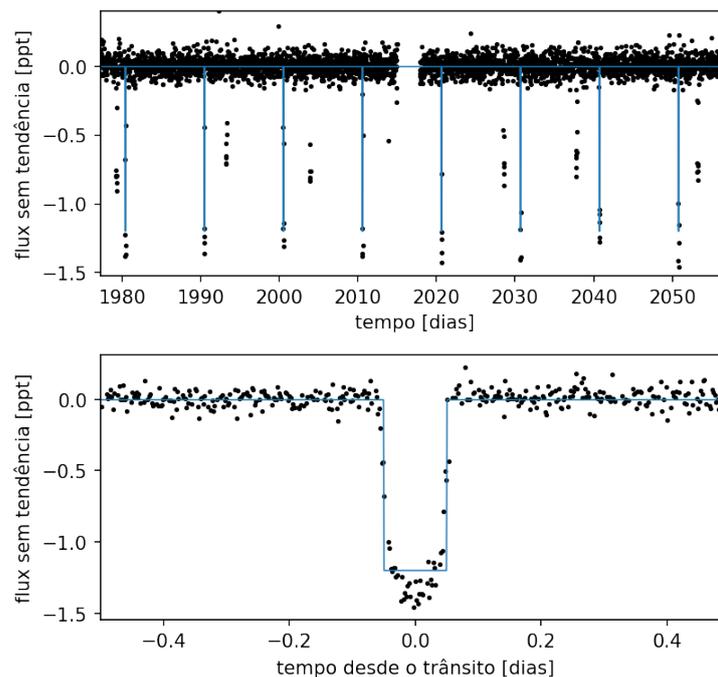


Figura 10: Exemplo de curva de luz extraído durante o desenvolvimento do atual estudo.

As curvas de luz possuem informações relevantes para esse projeto informações como SAP_{FLUX} , $PDCSAP_{FLUX}$ e $FLUX$, que serão utilizados mais a frente.

O SAP_{FLUX} é o fluxo em unidades de elétrons por segundo contido no ótimo pixels de abertura coletados pela espaçonave [26]. Já o $PDCSAP_{FLUX}$ é o fluxo contido na abertura ideal em elétrons por segundo após o módulo PDC ter aplicado seu algoritmo de cotendência à curva de luz [26]. Geralmente esse tipo de informação é mais limpo que o SAP_{FLUX} [27]. Por sua vez o $FLUX$ é uma imagem por cadência do fluxo medido em cada pixel após o processamento, a remoção do fundo interpolado e a remoção de raios cósmicos. As unidades são elétrons por segundo [26].

d Ruídos Instrumentais

O telescópio espacial Kepler funciona usando dispositivos de carga acoplada. Ele coletou fótons e os processou como medições de fluxo. A matriz de detectores consistia em 25 módulos e 94,6 milhões de pixels. Com um instrumento deste tamanho, os problemas de ruído eletrônico são inevitáveis [28]. Todos os dados abaixo foram extraídos da documentação do LightKurve™. Note que apenas alguns ruídos foram incluídos, devido a finalidade do estudo.

1 Crosstalk

Um desses eventos é o crosstalk. Crosstalk é o fenômeno no qual os sinais eletrônicos viajam de um canal de um dispositivo para outro e criam algum ruído.

Nos dados Kepler e K2, isso geralmente aparece como um sinal de um alvo (geralmente brilhante) que aparece em outro lugar do detector. Isso é chamado de "crosstalk de vídeo" e devido à blindagem entre os módulos, só tem um impacto significativo se ocorrer no mesmo módulo. [28].

No entanto, esses tipos de problemas são bem compreendidos e resolvidos pelo pipeline antes que os dados sejam entregues.

2 Bandas Rolantes

A eletrônica do detector Kepler é caracterizada por ruído de alta frequência devido às ressonâncias do circuito na faixa de GHz, que é transferido para frequências mais baixas através da taxa de amostragem de pixels da leitura do detector. Esses problemas causam alterações na temperatura e na sensibilidade dos pixels no plano focal, que aparecem como faixas horizontais móveis no detector. [28].

Os níveis de fluxo de fundo dos dados de pixels do Kepler não são estáticos. Em particular, alguns canais CCD ocasionalmente encontram um efeito de "banda rolante", onde o fundo tem um forte componente variável no tempo e a "banda" se move para cima através do detector. [28].

As bandas rolantes muitas vezes adicionam sinais espúrios aos dados que parecem mudanças astrofísicas reais. A melhor forma de posicionar a banda de rotação é variar o tamanho da abertura. Se a intensidade do sinal aumentar à medida que você aumenta o número de pixels de fundo na abertura, o efeito provavelmente será um componente aditivo de fundo. [28].

3 Raios Cósmicos e Queda Repentina de Sensibilidade de Pixel

Quando um raio cósmico atinge um pixel em um detector, ele aumenta imediatamente a leitura do fluxo desse pixel, injetando ruído entre as leituras.

A queda repentina de sensibilidade de pixel (SPSD) ocorre quando os raios cósmicos reduzem a sensibilidade de um grupo de pixels que atingem. Os eventos SPSP são classificados como danos de médio a longo prazo (ou permanentes). Durante eventos de médio prazo, os pixels afetados retornarão gradualmente à sensibilidade pré-SPSP. Danos a longo prazo exigiriam recozimento para serem corrigidos, algo que o Kepler não pode fazer, portanto equivale essencialmente a danos permanentes. Pixels afetados por danos permanentes ou de longo prazo mantiveram sensibilidade mais baixa durante toda a missão. [28]

4 Downlink em Dados

Durante a missão Kepler, o telescópio girará uma vez por mês para apontar para a Terra e correlacionar dados de meses anteriores. Isso deixa lacunas nos dados da série temporal porque o telescópio não pode coletar durante o downlink. Quando o telescópio retorna a um ponto fixo, seu movimento causa as chamadas "transições térmicas". Isto significa que ocorrem mudanças de temperatura nos componentes eletrônicos do telescópio e do detector, causando um aumento temporário nas leituras da contagem de elétrons. Mudanças na temperatura podem causar pequenas alterações no foco do seu telescópio. Isto aparece nos dados de fotometria de abertura única (SAP) como uma inclinação descendente causada pelo "reaquecimento" à medida que o fluxo retorna ao seu nível anterior. [28]

5 Argabrightening

Argabrightening é um problema de qualidade de cadência única que se parece com um raio cósmico no detector. Na verdade, acredita-se que o Argabrightening seja causado por detritos que atingem o

instrumento, causando um breve aumento no fluxo. Isto é diferente do evento eletrônico causado por um raio cósmico que corrompe a leitura do pixel: o brilho arga é o resultado da iluminação física. [28]

6 Reação das rodas

A atitude do Kepler foi controlada inicialmente por quatro rodas de reação. [28]

Os cruzamentos ocorrem quando as rodas de reação têm velocidade angular zero. Isso fez com que a ponta do telescópio se degradasse por alguns minutos de cada vez. Devido a esta curta escala de tempo, os cruzamentos da roda de reação são principalmente um problema nos dados de cadência curta, onde se manifestam como picos negativos nos dados de fluxo. Os eventos de cruzamento zero tornaram-se menos proeminentes após a falha de uma das rodas de reação do Kepler no Trimestre 14, devido a um aumento na velocidade das rodas de reação restantes. Não houve cruzamentos zero da roda de reação na missão K2. [28]

A dessaturação do momento foi uma consequência de um aumento de torque nas rodas de reação. Eventos de dessaturação ocorreram a cada 146 cadências longas durante a missão Kepler, levando ao modo de apontamento grosseiro e valores NaN nas curvas de luz para uma cadência longa ou várias cadências curtas por vez. [28]

e Ruídos de Imagem

O ruído de fótons, também conhecido como ruído de Poisson, é uma forma básica de incerteza associada à medição da luz, inerente à natureza quantizada da luz e à independência das detecções de fótons.

Sua magnitude esperada depende do sinal e constitui a fonte dominante de ruído na imagem, exceto em condições de pouca luz. Os sensores de imagem de fundo medem a irradiância da cena contando o número de fótons discretos incidentes no sensor durante um determinado intervalo de tempo. Nos sensores digitais, o efeito fotoelétrico é usado para converter fótons em elétrons, enquanto os sensores baseados em filme dependem de reações químicas fotossensíveis. Em ambos os casos, a independência das chegadas aleatórias de fótons individuais leva ao ruído de fótons, uma forma de incerteza dependente do sinal que é uma propriedade do próprio sinal subjacente. [29]

Na visão computacional, uma aproximação generalizada é modelar o ruído da imagem como independente do sinal, geralmente usando um gaussiano aditivo de média zero. Embora este modelo simples seja suficiente para algumas aplicações, é fisicamente irrealista. Em sistemas de imagem reais, o ruído de fótons e outras fontes de ruído baseadas em sensores contribuem em proporções variadas em diferentes níveis de sinal, levando a um ruído que depende do brilho da cena. Compreender o ruído do fóton e modelá-lo explicitamente é especialmente importante para tarefas de visão computacional de baixo nível que tratam imagens ruidosas e para a análise de sistemas de imagem que consideram diferentes níveis de exposição ou ganhos de sensor. [29]

4 ESTADO DA ARTE: "O ARTISTA"

a Sistemas baseado em Modelos

Os sistemas baseados em modelos são uma classe de algoritmos e abordagens que dependem de modelos explícitos matemáticos do problema que estão tentando controlar ou prever. Esses modelos descrevem as relações entre os diferentes componentes do sistema por meio de equações e são normalmente construídos com base em conhecimentos ou observações anteriores. São eles partes essenciais das teorias ou modelos científicos que tendem a representar a relação entre o mundo real e a matemática [30]. Esse tipo de modelo ao fazer classificações usa representações do ambiente, expectativas e cálculos prospectivos para fazer previsões classificativas de valor futuro [31].

Como citado na seção acima, com as análises feitas por Kepler a partir de suas observações de que a velocidade orbital dos planetas em torno do Sol não era constante, surgiram as leis do movimento planetário que nada mais é que uma representação matemática do que foi observado, ou seja, um modelo matemático.

Foi com tal observações que os cientistas atuais foram capazes de determinar o tamanho e o período orbital dos exoplanetas analisados, podendo assim determiná-los como habitável ou não. Com eles e a análise de seus campos gravitacionais é possível prever as órbitas de satélites e naves espaciais. Esta informação é crucial para manter as espaçonaves em órbitas desejadas, como a órbita do Sol no caso do telescópio.

b Sistemas sem Modelo

Os Sistemas sem Modelo, *Model Free*, referem-se a algoritmos que não dependem de um modelo explícito matemático para tentar que controlar ou prever uma informação específica. Em vez disso, eles aprendem diretamente com os dados, normalmente por meio de tentativa e erro [31].

Os pontos fortes dos Sistemas sem Modelos estão particularmente em domínios onde a modelagem explícita é impraticável. Isso ocorre na robótica, por exemplo, uma vez que a dinâmica do mundo real pode ser complexa e difícil de capturar com precisão. Os métodos sem modelo permitem que os robôs se adaptem e aprendam com as suas interações com o ambiente, permitindo-lhes realizar tarefas com um elevado grau de autonomia.

Os Sistemas sem Modelos têm um papel significativo a desempenhar no campo da Inteligência Artificial permitindo que as máquinas aprendam e tomem decisões sem depender de modelos explícitos do sistema ou ambiente subjacente, como será explicado a seguir.

c Sistemas Inteligentes

Os sistemas inteligentes são referidos como sistemas que incorporam inteligência artificial para realizar tarefas de maneira autônoma ou parcialmente autônoma. Eles são capazes de processar informações, aprender e tomar decisões.

A Inteligência Artificial, mais comumente referenciada como IA, surgiu como um dos campos mais transformadores da ciência da computação, remodelando a forma como o ser humano interage com a tecnologia. Ela transformou sonhos em realidade, trazendo a magia dos cinemas para a mesa do consumidor. Um exemplo, é a Siri, assistente virtual da Apple. De acordo com os fundadores e engenheiros da Apple, a Siri foi inspirada no sistema de IA HAL 9000 do filme "2001: Uma Odisseia no Espaço" de Stanley Kubrick [32]. O filme, lançado em 1968, apresentava HAL como uma inteligência artificial com avançada compreensão de linguagem natural e capacidade de comunicação com humanos bem similar a Siri.



Figura 11: Representação do HAL 9000 e a Siri.

A IA abrange o desenvolvimento de sistemas computacionais capazes de executar tarefas que normalmente requerem inteligência humana como percepção visual, reconhecimento de fala, aprendizagem e tomada de decisões. Esta área de estudo tem o potencial de revolucionar inúmeras áreas de conhecimento, desde a saúde a exploração de planetas.

Trazendo um pouco para a pesquisa em discussão, a Inteligência Artificial, pode auxiliar no planejamento de missões espaciais, incluindo a seleção de locais de pouso, como analisar grandes quantidades de dados coletados, extraindo informações valiosas e identificando padrões, como será realizado nesse estudo [33].

A ideia da existência de seres artificiais com capacidades semelhantes às humanas remonta à mitologia antiga, com histórias de autômatos e golens. Porém a IA realmente começou a surgir nos meados do século XX, quando pioneiros como Alan Turing e John McCarthy lançaram as bases teóricas. Os primeiros esforços concentraram-se em regras e representações explícitas para modelar processos cognitivos humanos. Contudo, esta abordagem encontrou limitações ao tratar problemas mais complexos, o que levou ao começo da "aprendizado de máquina".

O aprendizado de máquina, conhecido mundialmente como *Machine Learning*, permite que a partir de uma base de dados os sistemas aprendam e melhorem seu desempenho ao longo do tempo. Nele, emprega-se algoritmos que permitem que os computadores identifiquem padrões, façam previsões e até otimizem processos de tomada de decisão [34].

Aprofundando um pouco mais, o primeiro passo na construção de um sistema inteligente é coletar dados relevantes que servem de base para o treinamento do modelo. A quantidade e a qualidade dos dados são cruciais para a precisão dele. Depois, é necessário tratar eles, visto que na sua forma bruta podem ser confusos para análise. Essa parte pode envolver remoção de duplicatas, tratamento de valores ausentes e conversão de dados em um formato adequado para processamento. Com isso realizado, é necessário atribuir recursos ou características específicas aos dados que o modelo usa para fazer previsões ou decisões. A seleção ou extração eficaz de recursos é crítica para o desempenho do modelo.

O próximo passo é a escolha do tipo apropriado de modelo de Inteligência Artificial. Os tipos comuns incluem Rede Neural, Lógica Fuzzy entre outros, para isso é necessário entender o tipo de problema que deseja abordar e os tipos de informações que se tem sobre.

Com o modelo definido inicia-se a fase de treinamento, na qual o modelo é alimentado com uma grande quantidade de dados rotulados fazendo com que ele aprenda as relações de entrada e saída desejada. O modelo auto ajusta seus parâmetros internos para minimizar erros de previsão. Após o treinamento, o sistema é testado introduzindo um conjunto separado de dados que nunca foi visto antes. Isso ajuda a avaliar seu desempenho de generalização e a identificar possíveis problemas como overfitting, onde a IA memoriza os dados de treinamento, mas não consegue generalizar para novos dados.

A seguir serão introduzidos três tipos de modelos de Aprendizado de Máquina que podem ser escolhidos para futuros projetos.

1 Rede Neural

Em 1943, Warren McCulloch e Walter Pitts introduziram um modelo matemático de um neurônio simplificado que representou o início para o estudo do campo de redes neurais artificiais [35]. Hoje, elas são uma tecnologia fundamental no campo da inteligência artificial e são utilizadas numa ampla gama de aplicações, desde cuidados de saúde a veículos autônomos.

Trazendo um pouco para o estudo atual, as redes neurais são extremamente usadas para exploração do espaço, visto que com elas é possível processar e analisar dados de telescópios espaciais para identificar e classificar exoplanetas. Eles ajudam a distinguir potenciais candidatos, o que é o que esse estudo propõem.

As redes neurais, por fim, são uma classe de modelos de aprendizado de máquina inspirados na estrutura e no funcionamento do cérebro humano, onde o objetivo não é replicar inteiramente o cérebro, mas aproveitar alguns de seus princípios. De modo geral, as redes neurais são uma série de algoritmos projetados para reconhecer padrões nos dados, tornando-os capazes de realizar tarefas que antes eram consideradas complexas demais para serem executadas pelos computadores [36].

A Rede Neural é inspirada no cérebro humano [34]. Em um cérebro, os neurônios são os blocos básicos responsáveis pelo processamento e transmissão de informações. Numa rede neural, os neurônios são artificiais, muitas vezes chamados de nós. Cada um deles recebe múltiplas entradas, vindos da base de dados. Essa parte inicial de introdução de informações é chamada de camada de entrada.

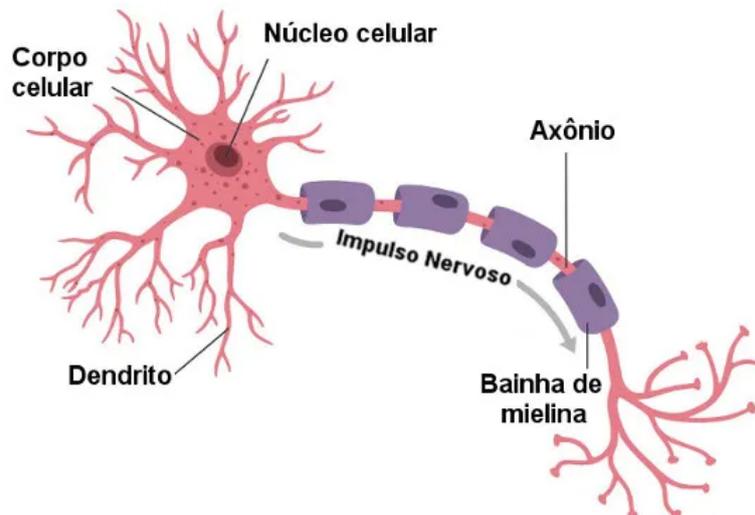


Figura 12: Representação visual de um neurônio humano. [7]

Assim que a entrada recebe um dados ela passa por uma ou mais camadas ocultas. Nessas camadas ocultas, são realizados sob os neurônios artificiais cálculos matemáticos, que na verdade são introdução de pesos e bias, que determinam a força de conexão entre nós.

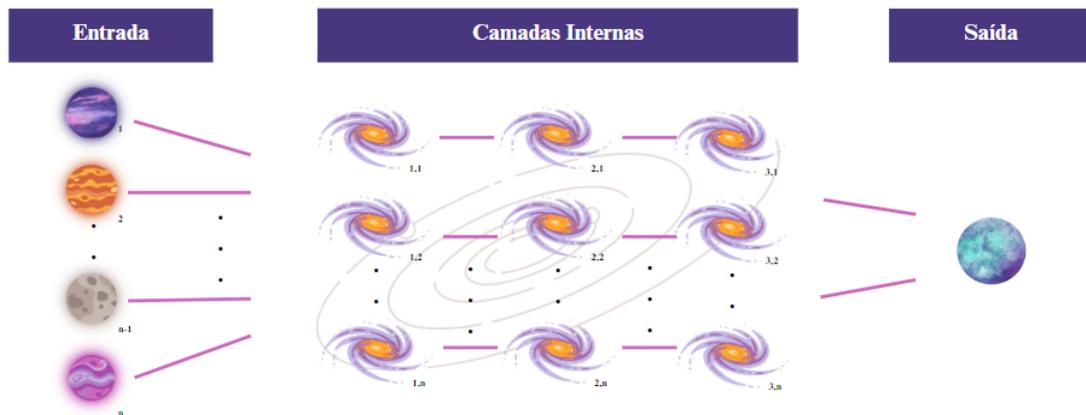


Figura 13: Exemplo de Rede Neural com três camadas internas.

Os Pesos e os bias são os parâmetros que a rede aprende durante o seu treinamento. Os pesos são aplicados às entradas de cada nó e seus valores são ajustados por meio de um processo denominado retropropagação durante o treinamento. Os bias são adicionados à soma ponderada antes de passar por uma função de ativação para produzir a saída do neurônio.

As funções de ativação introduzem não linearidade no modelo, permitindo que as redes neurais aprendam relacionamentos complexos. Funções de ativação populares incluem sigmóide, ReLU (Rectified Linear Unit) e outras.

A rede neural tem duas fases: o treinamento e o teste ou validação. Durante o treinamento, a rede é alimentada com um conjunto de dados contendo pares de entrada-saída. Os pesos e tendências são atualizados iterativamente através de muitas passagens pelos dados até que o modelo aprenda a fazer previsões precisas. Após o treinamento, o modelo é avaliado em um conjunto de testes separado para avaliar seu desempenho em dados não vistos. Uma vez treinado, o modelo pode ser usado para fazer previsões sobre dados novos e não vistos.

Em seguida, é calculada a função de perda que mede quão bem as previsões da rede neural correspondem aos valores alvo reais. Ela fornece uma medida quantificável do desempenho do modelo. Dependendo desta função, é necessário reequilibrar os pesos.

Embora as redes neurais sejam essenciais no dia-a-dia, principalmente com a propagação do CHATGPT™, os desafios persistem. Para desenvolvimentos mais complexos é necessário recursos computacionais mais robustos, o que tornam os projetos mais caros. Além disso, considerações éticas e transparência são fundamentais, garantindo que estes modelos poderosos sejam implementados de forma responsável [34].

Contudo, projetos cinematográficos não deixam de explorar futuras possibilidades para o modelo. "JARVIS" (*"Just A Rather Very Intelligent System"*) é o nome do assistente virtual fictício no Universo Marvel, conhecido por auxiliar o Homem de Ferro. No contexto da ficção, J.A.R.V.I.S. é uma inteligência artificial avançada e altamente sofisticada que possui uma ampla gama de habilidades, incluindo reconhecimento de voz, aprendizado de máquina e processamento de linguagem natural como HAL 9000. Além disso, esse sistema é capaz de aprender e adaptar suas respostas com base em interações com os usuários de forma eficiente.

d Tipos de Redes Neurais

1 Single Layer Perceptron

O Perceptron é uma unidade fundamental na arquitetura de redes neurais, inspirada na estrutura e funcionamento de um neurônio biológico. Foi desenvolvido por Frank Rosenblatt no final da década de 1950 e serve como base para modelos de redes neurais mais complexos [35].

O Single Layer Perceptron é um algoritmo de aprendizagem supervisionada que classifica os dados em duas categorias, ou seja, classificador binário. Ele é considerado a menor unidade de rede neural que

calcula uma soma ponderada das entradas, subtrai um limite e passa um dos dois valores possíveis como resultado [35]. Por ser, um classificador binário, o Perceptron pode ser usado para implementar portas lógicas, porém ele aprende apenas problemas linearmente separáveis.

O Perceptron aprende por meio de um processo de ajuste de seus pesos com base no erro de suas previsões. Isso é conseguido usando um algoritmo de aprendizagem conhecido como regra de aprendizagem Perceptron. Se o Perceptron classificar incorretamente uma entrada, ele atualizará seus pesos para minimizar o erro.

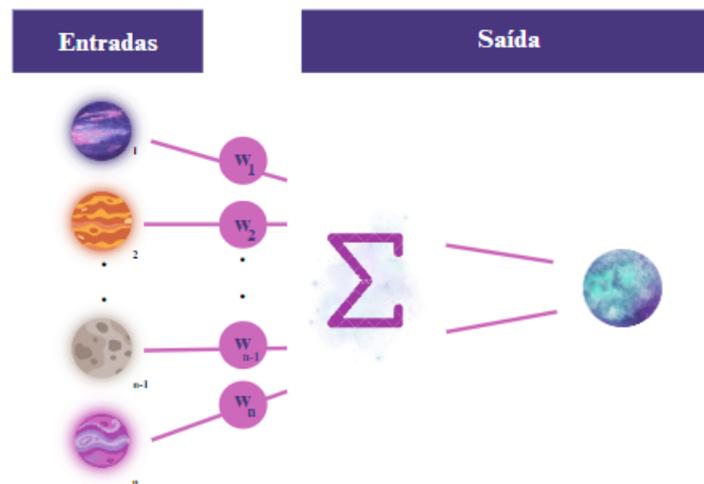


Figura 14: Representação visual de um Perceptron.

Uma das principais limitações de um Perceptron de camada única é sua incapacidade de lidar com dados separáveis não linearmente. Ele só pode aprender limites de decisão lineares. Isso levou ao desenvolvimento de Perceptrons multicamadas (MLPs) com camadas ocultas, que podem aprender padrões mais complexos [37].

Trazendo a explicação para o mundo cinematográfico, o filme *Divertidamente*™ da Pixar™, se passa na mente de uma menina chamada Riley, focando nas emoções que a guiam em sua vida diária. Na central das emoções há uma classificação de memórias. No início do filme as emoções são classificadas entre memória importante ou descartável. Cada memória é ponderada em sua importância a partir do que a personagem "Alegria" acha essencial para o desenvolvimento da jovem. No caso do Single Layer Perceptron ou a memória será importante para a personagem ou não, e de acordo com o peso da opinião da "Alegria" as classificações serão ajustadas futuramente. Como a "Alegria" é só uma e ela possui um bias, logo é necessário um modelo mais complexo, com mais feedback dos outros personagens é necessário.

Apesar de sua simplicidade, os Single Layer Perceptrons encontraram aplicações em áreas como tarefas de classificação binária, operações lógicas e problemas simples linearmente separáveis. Na prática, eles são frequentemente usados como blocos de construção em arquiteturas de redes neurais mais complexas.

O Perceptron serve como um conceito fundamental na teoria das redes neurais, demonstrando como uma unidade computacional simples pode tomar decisões com base em entradas ponderadas. Embora limitado em suas capacidades, abriu caminho para o desenvolvimento de modelos de redes neurais mais sofisticados, capazes de lidar com tarefas complexas.

2 MultiLayer Perceptron

Como dito acima, redes de uma camada não são capazes de solucionar problemas que não sejam linearmente separáveis. Com o desenvolvimento do trabalho de Rumelhart, Hinton e Williams em 1986, foi desenvolvido um modelo capaz de treinar eficientemente redes com camadas intermediárias, resultando no modelo de Redes Neurais Artificiais mais utilizado atualmente, as redes Perceptron Multi-Camadas (MLP), treinadas com o algoritmo backpropagation [37].

O Multilayer Perceptron é usado para classificações mais complexas. Nele temos uma propagação bidirecional (*backpropagation*), ou seja, propagação para frente e propagação para trás. Os dados de entrada viajam através de várias camadas de neurônios artificiais, onde cada nó está conectado a todos os neurônios da próxima camada, obtendo assim uma rede neural totalmente conectada [37].

Essas entradas são multiplicadas por pesos e alimentadas para a função de ativação e na retropropagação são modificadas para reduzir a perda. Os pesos são nada mais nada menos que valores aprendidos por máquina em redes neurais. Eles se ajustam automaticamente dependendo da diferença entre os resultados previstos e as entradas de treinamento.

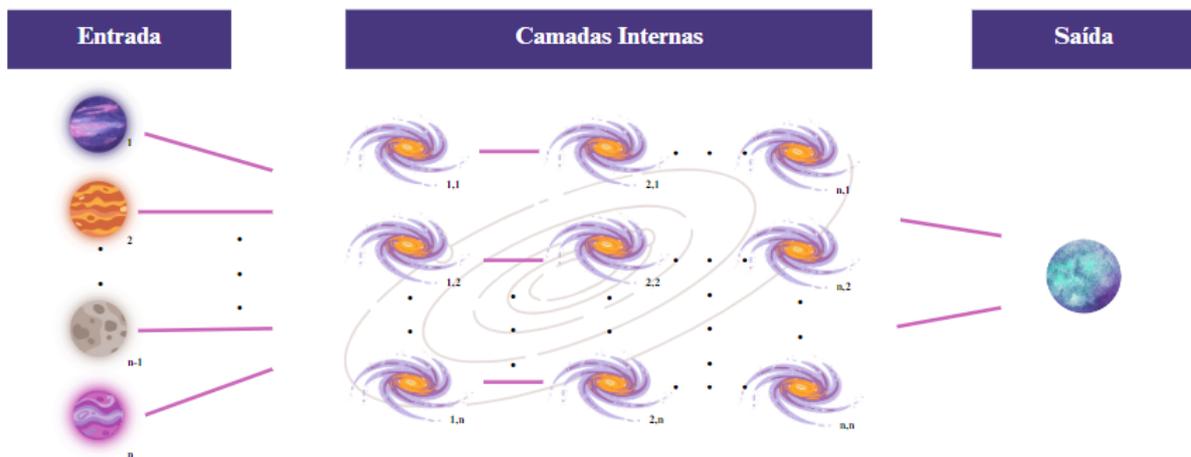


Figura 15: Representação visual de um MultiLayer Perceptron com três camadas.

Os MultiLayer Perceptron são altamente versáteis e encontram aplicação em uma ampla variedade de campos, incluindo visão computacional (por exemplo, reconhecimento de imagem com redes neurais convolucionais), processamento de linguagem natural, análise de séries temporais e muito mais. Eles se destacam em tarefas onde existem relacionamentos complexos dentro dos dados.

As MLPs servem como base para arquiteturas de redes neurais mais avançadas, como redes neurais profundas. Eles consistem em muitas camadas ocultas, permitindo representação e abstração de recursos ainda mais complexas.

3 Single Layer Perceptron vs. MultiLayer Perceptron

O Single Layer Perceptron (SLP) e o Multilayer Perceptron (MLP) mesmo sendo dois tipos de redes neurais, elas apresentam diferenças significativas entre suas arquiteturas e capacidades. Na prática, os MLPs são muito mais versáteis e amplamente utilizados devido à sua capacidade de lidar com uma gama mais ampla de tarefas.

O primeiro ponto que demonstra a diferença é que SLP é mais simples e limitado a problemas linearmente separáveis, já o MLP é mais complexo e capaz de lidar com relacionamentos não lineares.

Além disso, é importante destacar as diferenças do número de camadas e das funções de ativação. O SLP possui apenas uma camada e normalmente usa funções lineares ou escalonadas, já o MLP possui múltiplas camadas, incluindo pelo menos uma camada oculta e usa funções de ativação não lineares nas camadas ocultas. Mais camadas ou mais neurônios não significa necessariamente que você obterá melhor resultado/desempenho na rede. Se seus dados forem muito simples ou o número de observações não for tão alto, adicionar mais parâmetros (mais camadas/neurônios) poderá resultar em ajuste excessivo dos dados *overfitting*.

O *overfitting* ocorre quando a sua rede não consegue generalizar bem, ou seja, para um grupo de entradas ela tem uma boa acurácia, ou seja, um bom desempenho. Porém ao introduzir novos dados, o sistema não consegue generalizar a resposta. Um exemplo é uma base de dados com informações de temperatura em graus Kelvin (K) para saber se o planeta é habitável ou não. Se em primeira instância fizer um treinamento

da rede com apenas essa temperatura e depois fizer uma análise com dados em graus Celsius ($^{\circ}\text{C}$) ou Fahrenheit ($^{\circ}\text{F}$), um sistema não vai entender o outro e conseguir classificar direito.

Esse erro é muito comum na montagem de redes neurais, pois quando é feito o primeiro treinamento das previsões geralmente é utilizado um conjunto de dados conhecido. Logo, é necessário após testes, alimentar a rede com outra base de dados para ver se em casos atípicos ela consegue generalizar.

Além disso, para o SLP, independente da função de ativação a ser utilizada, tem-se apenas duas possibilidades de valores a serem produzidos pela sua saída, ou seja, valor 0 ou 1. [38] Dessa forma, valores de entrada são apresentados ao perceptron, e se a saída esperada for igual à desejada, então o desempenho é considerado satisfatório e não é necessário fazer nenhuma alteração nos pesos. No entanto, se a saída não é igual à desejada, os pesos precisarão ser alterados para reduzir o erro. [39] Tal processo é muito utilizado em reconhecimento de padrões.

Outro ponto a comentar é que se for utilizado uma não-linearidade de saturação como função de ativação (ou seja, sigmóides ou tanh), que serão discutidas a frente, e tentar adicionar muitas camadas poderá resultar com que a rede treine muito lentamente, ou talvez nem sequer treine.

Comentando um pouco das vantagens de usar um MLP em relação aos outros tipos de redes neurais, pode-se dizer que seu ponto forte é a capacidade de modelar relações não lineares complexas entre as entradas e as saídas. Com ele é possível realizar tarefas como reconhecimento de fala e imagem, processamento de linguagem natural além de séries temporais, devido à sua capacidade de aproximar qualquer função, visto que possui camadas e neurônios suficientes. [38]

Outra vantagem do MLP é que utiliza uma abordagem de aprendizagem supervisionada, ou seja, é relativamente fácil de treinar. O algoritmo de retropropagação, comumente usado para treinar MLPs, usa um método para ajustar os pesos da rede a fim de minimizar o erro entre a saída prevista e a saída verdadeira. [38] O MLP também tende a ser mais robusto ao ruído/generalizações do que outros tipos de redes neurais, como os SLP.

4 Lógica Fuzzy

A lógica fuzzy foi introduzida por Lotfi A. Zadeh em um artigo seminal intitulado "Conjuntos Fuzzy" publicado em 1965. [40] Zadeh propôs uma nova abordagem para lidar com a incerteza, introduzindo o conceito de "conjuntos fuzzy" onde os elementos têm graus de pertinência. Suas aplicações abrangem setores e contribuíram para avanços em sistemas de controle, automação e suporte a decisões [41].

Trazendo um pouco mais para o estudo atual, a lógica fuzzy tem sido aplicada em sistemas de apoio à decisão para lidar com dados imprecisos e ambientes incertos, como é o caso da exploração de exoplanetas, visto que é necessário saber se uma observação do telescópio é um exoplaneta ou não baseado na imprecisão dos dados.

Como dito acima, a lógica fuzzy é uma estrutura matemática que trata do raciocínio e da tomada de decisões em situações de incerteza e imprecisão. Ao contrário da lógica binária tradicional, que opera com valores estritos de verdadeiro ou falso, a lógica fuzzy permite valores intermediários, permitindo uma abordagem mais sutil para a resolução de problemas.

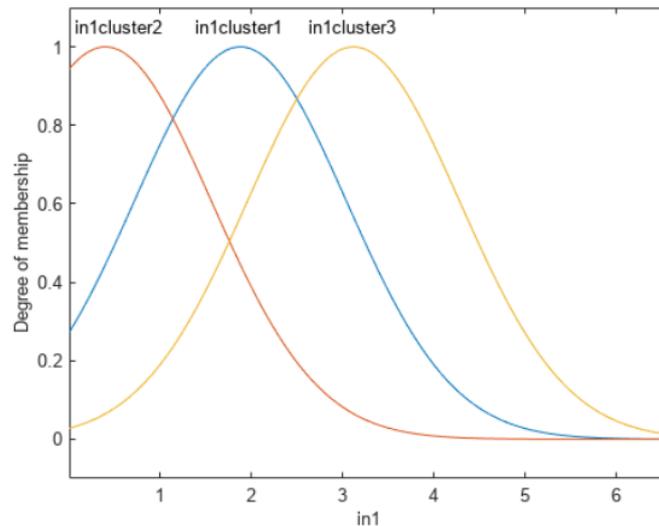


Figura 16: Exemplo de Funções de Pertinência [8].

No centro da lógica fuzzy está o conceito de conjuntos fuzzy. Um conjunto fuzzy é uma coleção de elementos com graus de pertinência variando de 0 a 1, indicando até que ponto cada elemento pertence ao conjunto. Isto contrasta com a teoria clássica dos conjuntos, onde a pertinência é estritamente 0 ou 1. Logo, eles fornecem uma maneira natural de representar e raciocinar sobre informações não binárias [42].

Um dos pontos fortes da lógica fuzzy reside na sua capacidade de lidar com variáveis linguísticas. Estas são variáveis cujos valores são descritos usando termos de linguagem natural, como "triste", "feliz", "nojo", "medo" ou "raiva". Na lógica tradicional, é um desafio definir com precisão os limites de tais termos. Esse modelo, no entanto, permite atribuir graus de pertinência a estes valores linguísticos, fornecendo pequenas nuances a classificações. [42].

Trazendi a análise para o mundo do cinema, em Divertidamente™, além de classificar as memórias entre importante e descartável, elas são classificadas também pela emoção que representa. No início do filme cada uma delas era apenas uma emoção, já no final elas se mesclam, mostrando que nem sempre os limites em situações de análise são tão claros e definidos.

Já na análise de O Pequeno Príncipe™ a lógica fuzzy pode ser aplicada na hora categorizar as lições do livro. Ao invés de defini-las como estritamente "certas" ou "erradas", a lógica fuzzy pode considerar o contexto das situações. Por exemplo, a lição de lidar logo com o baobá pode ser avaliada em uma escala que considera a complexidade da situação no momento e o quanto a árvore já cresceu (o que é grande ou pequeno?).

Por fim, as regras fuzzy são os blocos de construção do sistema. Essas regras assumem a forma de declarações "se-então", onde a parte "se" consiste em condições expressas em termos de conjuntos fuzzy, e a parte "então" define a ação ou conclusão. Se existirem múltiplas regras fuzzy aplicáveis a uma determinada situação, as suas conclusões precisam ser combinadas. Isso normalmente é feito usando operadores max e min [42].

Após a agregação de regras, o conjunto de saída fuzzy precisa ser convertido em um valor de saída exato. Logo, ocorre a defuzzificação, que é o processo de encontrar o valor mais representativo do conjunto fuzzy. Isso geralmente é feito calculando o centróide do conjunto de saída fuzzy.

5 Rede Neuro-Fuzzy

A interseção entre a rede neural e a lógica fuzzy resultou na criação do campo de estudo conhecido como redes neuro-fuzzy. Esta abordagem híbrida combina a capacidade de aprendizado das redes neurais com a capacidade de representação de dados incertos da lógica fuzzy [43].

Como dito acima, as redes neuro-fuzzy são sistemas baseados em regras que integram elementos de aprendizado de máquina com a lógica fuzzy, modelos híbrido [43]. Elas possuem a capacidade de aprender com dados e adaptar-se a novos contextos, enquanto lidam de forma eficiente com informações imprecisas e incertas. A estrutura da rede neuro-fuzzy é composta por camadas que se assemelham às camadas de uma rede neural tradicional com suas camadas de entrada, ocultas e de saída. A principal diferença está na utilização de funções de pertinência e operações de inferência baseadas em lógica fuzzy, permitindo a manipulação de conceitos vagos e incertos.

Uma das principais vantagens das redes neuro-fuzzy é a sua capacidade de modelar sistemas complexos sem a necessidade de um entendimento profundo das relações subjacentes. Isso significa que elas podem ser aplicadas em situações onde o modelo matemático é desconhecido. Além disso, as redes neuro-fuzzy são robustas o suficiente para lidar com dados ruidosos e incompletos, tornando-as valiosas em ambientes reais, onde a incerteza é inevitável.

As redes neuro-fuzzy têm encontrado uma ampla gama de aplicações em diversos setores. Na indústria, elas são utilizadas para otimizar processos de produção, realizar diagnósticos de falhas em sistemas complexos e controlar sistemas dinâmicos não-lineares. No campo da medicina, as redes neuro-fuzzy têm sido empregadas para a tomada de decisões em diagnósticos médicos, previsão de doenças e modelagem de sistemas biológicos. Além disso, elas têm demonstrado eficácia em sistemas de controle de tráfego, previsão meteorológica e análise financeira. [43].

Apesar das inúmeras vantagens, as redes neuro-fuzzy não estão isentas de desafios. A determinação adequada da arquitetura e dos parâmetros do modelo pode ser uma tarefa complexa e exigir expertise considerável. Além disso, o processo de treinamento pode ser computacionalmente intensivo, especialmente em conjuntos de dados grandes e complexos. É importante também destacar que, como qualquer abordagem baseada em dados, a qualidade e a representatividade dos dados utilizados no treinamento são cruciais para a eficácia do modelo.

6 ANFIS

Uma técnica neuro-fuzzy chamada de sistema de inferência fuzzy baseado em rede adaptativa (ANFIS) é realizada com a fusão entre a rede neural e o sistema fuzzy de inferência. Os valores da lógica fuzzy levam em conta a imprecisão e a incerteza do sistema que está sendo modelado enquanto a rede neural dá uma sensação de adaptabilidade [44].

Usando este método híbrido, inicialmente um modelo fuzzy inicial junto com suas variáveis de entrada são derivadas com a ajuda das regras extraídas dos dados de entrada e saída do sistema que está sendo modelado. Em seguida, a rede neural é usada para ajustar as regras do modelo fuzzy para produzir o modelo ANFIS final do sistema. Neste trabalho proposto o ANFIS é usado como a espinha dorsal para a identificação de sistemas do mundo real [44].

e Métricas de avaliação de desempenho

Em aprendizado de máquina, a avaliação de desempenho de modelos de classificação é um passo crucial para determinar a eficácia e a confiabilidade de um algoritmo. Diversas métricas foram desenvolvidas para medir o desempenho desses modelos em diferentes contextos. Estas métricas oferecem uma visão detalhada sobre a capacidade de um modelo de lidar com dados de teste e são essenciais para a tomada de decisões informadas sobre a escolha e otimização de algoritmos de classificação.

- **Acurácia:**

É uma das métricas mais fundamentais para avaliar a performance de modelos de classificação. Ela representa a proporção de predições corretas em relação ao número total de predições. A acurácia pode ser danosa em casos de classes desbalanceadas, onde a prevalência de uma classe é muito maior que a outra. [45]

Ademais, ao comparar diferentes modelos, que é o caso do atual estudo, a acurácia pode ser uma referência interessante e rápida para determinar qual modelo está superando os outros em termos de precisão.

Sua fórmula é dada por:

$$Acurácia = \frac{\text{Total de predições corretas}}{\text{Total de predições feitas}} \quad (1)$$

Ou, pode ser calculada também:

$$Acurácia = \frac{VP + VN}{VP + VN + FN + FP} \quad (2)$$

Onde VP e VN correspondem a verdadeiro positivo e negativo respectivamente e FN e FP como falso negativo e positivo.

- **Recall e a Precisão:**

O Recall e a Precisão que são métricas que se concentram na capacidade do modelo de classificar corretamente instâncias da classe positiva. O Recall mede a proporção de instâncias corretamente identificadas em relação ao total de instâncias. Por outro lado, a Precisão representa a proporção de instâncias corretamente identificadas em relação ao total de instâncias classificadas como positivas pelo modelo. Essas métricas são especialmente importantes em situações onde a identificação correta da classe é crítica [45] [46].

Suas fórmulas são:

$$Recall = \frac{VP}{VP + FN} \quad (3)$$

$$Precisão = \frac{VP}{VP + FP} \quad (4)$$

- **F1-Score:**

O F1-Score é uma métrica que combina a Precisão e o Recall em uma única medida, fornecendo uma avaliação mais equilibrada do desempenho do modelo. Calculado como a média harmônica da Precisão e do Recall, o F1-Score é particularmente útil em cenários onde os falsos positivos e falsos negativos têm diferentes implicações e custos associados [45].

$$F1 - Score = 2 * \frac{Recall * Precisão}{Recall + Precisão} \quad (5)$$

- **RMSE:**

De acordo com Mario Filho, autor do "Manual Prático de Data Science" e Kaggle™Competitions Grandmaster: "A raiz do erro quadrático médio (RMSE) é calculada tomando-se a raiz quadrada da média dos quadrados dos erros, onde o erro bruto é a diferença entre o valor previsto pelo modelo e o valor real." [46]

A raiz do erro quadrático médio (RMSE) é calculada tomando-se a raiz quadrada da média dos quadrados dos erros, onde o erro bruto é a diferença entre o valor previsto pelo modelo e o valor real.

O RMSE é bom quando é desejado que o modelo foque em acertar valores extremos. Ele penaliza erros extremos mais do que erros pequenos. No entanto, isso torna-o sensível a estes valores extremos, o que pode distorcer a métrica. [46] Quanto menor o RMSE, melhor o desempenho.

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(Y_i - y_i)^2}{n}} \quad (6)$$

Onde Y_i é o valor previsto, y_i é o valor observado e n número de observações.

- **Matriz de Confusão:**

A matriz de confusão nada mais é que uma ferramenta visual que permite avaliar o desempenho de um modelo de aprendizado de máquina. Ela permite visualizar a distribuição das previsões feitas pelo modelo em relação aos casos reais. A partir da matriz de confusão, é possível calcular várias métricas citadas acima, como acurácia, precisão, recall e F1-Score. [47]

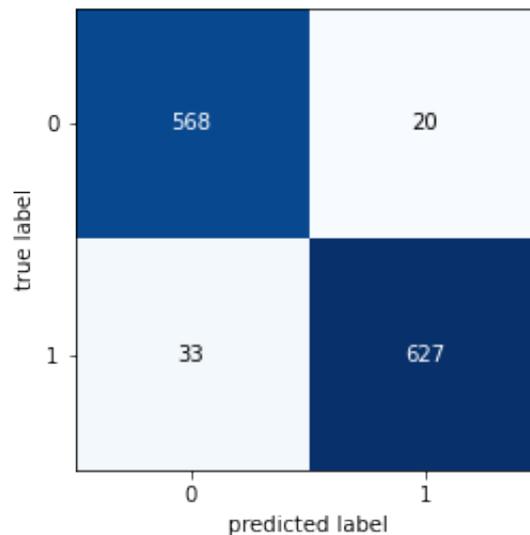


Figura 17: Representação de uma matriz de cofusão gerada no atual estudo. Nesta matriz foi utilizada um MLP de duas camadas e função de ativação RELU.

f Hiperparâmetros de Redes Neurais

Os hiperparâmetros são configurações ajustáveis que não são aprendidas a partir dos dados de treinamento, eles são definidos antes de inicializar o treinamento e afetam diretamente o comportamento final do modelo.

Para a construção de uma rede neural que possua um desempenho eficaz é necessário ter a configuração correta de seus hiperparâmetros. Experimentação e ajustes iterativos são muitas vezes necessários para encontrar os valores ideais para cada parâmetro em um determinado contexto.

Alguns hiperparâmetros são:

- **Taxa de Aprendizagem:**

A taxa de aprendizagem define a rapidez com que uma rede atualiza seus parâmetros. A baixa taxa de aprendizagem deixa o processo de aprendizagem mais lento, mas converge suavemente. Uma taxa de aprendizagem maior acelera a aprendizagem, mas pode não convergir. Normalmente, uma taxa de aprendizagem decrescente é preferida. [48]

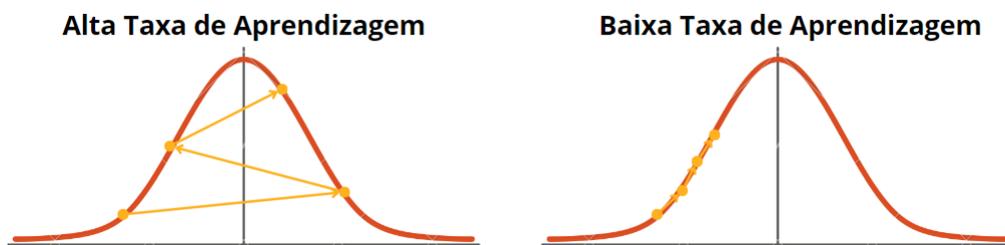


Figura 18: Visualização de funcionamento da taxa de aprendizado alto e baixo.

- **Número de Épocas:**

Define quantas vezes o algoritmo percorrerá o conjunto de dados de treinamento durante a fase de treinamento. [48]

- **Arquitetura da Rede:**

Inclui o número de camadas não ocultas, o número de neurônios em cada camada, o tipo de função de ativação, etc.

Como dito anteriormente, o número de camadas ocultas é um hiperparâmetro que pode impactar significativamente no desempenho do modelo. Ao aumentar as camadas ocultas da rede permite que ela aprenda relacionamentos mais complexos. Ainda assim, também pode aumentar o risco de *overfitting* e exigir mais recursos computacionais. [49]

O número de neurônios por camada oculta se comporta de forma semelhante ao número de camadas, quanto maior seu número maior o ajuste nos dados, por isso a sua escolha aumenta com a complexidade dos dados. Geralmente, o número de neurônios é aumentado até que não haja melhora no desempenho. É importante notar que este parâmetro também está sujeito a *overfitting*. [49]

- **Funções de Ativação:**

As funções de ativação são usadas para introduzir a não linearidade nos modelos, o que permite que eles aprendam os limites de previsão não linear. Exemplos de funções de ativação usadas no atual estudo são: Softmax, Log-Softmax, Tanh, HardTanh, ELU. [48]

De acordo com Pragati Baheti, engenheira de software da Microsoft™: "A função de ativação softmax é usada em redes neurais de classificação. Ela força a saída de uma rede neural a representar a probabilidade dos dados serem de uma das classes definidas." [50]

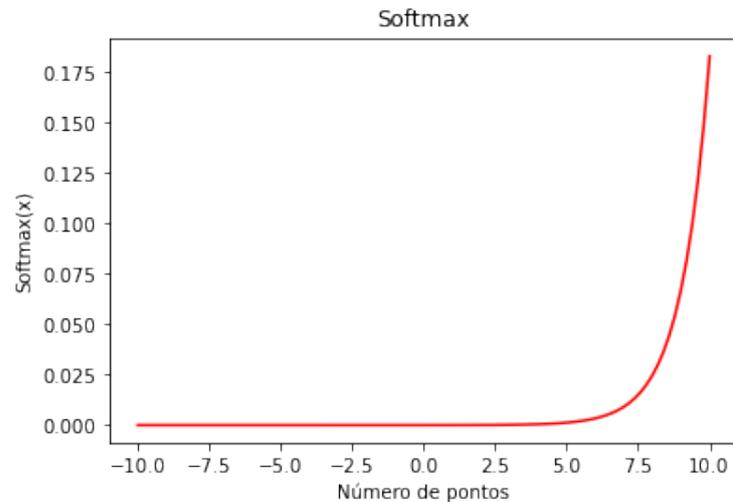


Figura 19: Visualização da função Softmax.

Já a função Log-Softmax usa o log de probabilidades sobre probabilidades. Ela tem melhor desempenho numérico que a Softmax. Essas vantagens podem ser extremamente importantes para a implementação, especialmente quando o treinamento de um modelo pode ser computacionalmente desafiador e caro. Em classificadores, ela penaliza fortemente o modelo quando ele falha em prever uma classe correta. [50]

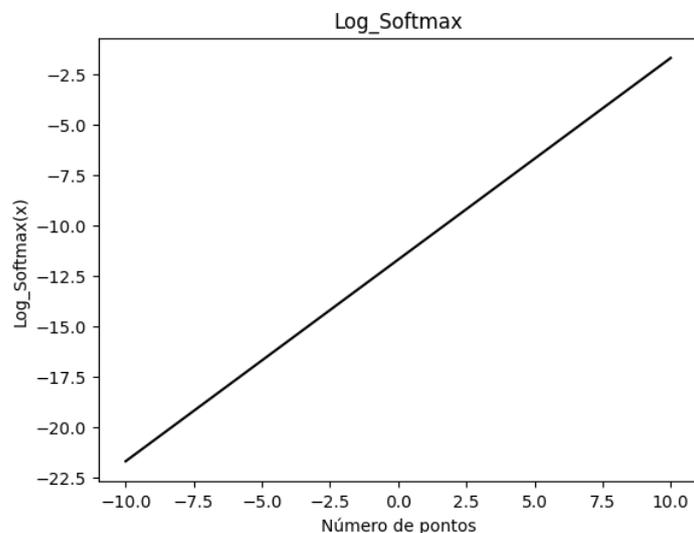


Figura 20: Visualização da função Log-Softmax.

Outra função de ativação comum é a função hiperbólica tangente, ou função Tanh. Essa função consegue trabalhar com estrutura de dados complexo e empurra os valores de entrada para 1 e -1 devido sua forma de S. Em Tanh, como a função é centralizada em zero, quanto maior a entrada (mais positiva), mais próximo o valor da saída estará de 1, enquanto quanto menor a entrada (mais negativa), mais próxima a saída estará de -1. Além disso, a média da camada oculta é 0 ou muito próxima dela. Isso ajuda a centralizar os dados e torna o aprendizado para a próxima camada muito mais fácil. [50]

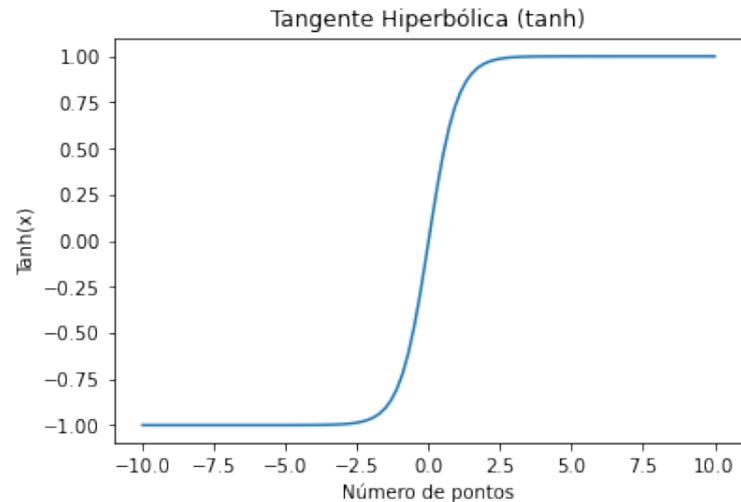


Figura 21: Visualização da função Tanh.

A função de ativação do Hardtanh pega qualquer valor de entrada e retorna um valor de -1 se a entrada for menor que -1, um valor de 1 se a entrada for maior que 1 e o próprio valor de entrada se estiver entre -1 e 1. É frequentemente usada em aprendizado de máquina para tarefas como processamento de linguagem natural [51] Vale comentar que um dos principais benefícios de usar a função de ativação do Hardtanh é que ela é menos complexa que outras funções de ativação, como tanh, logo ela requer menos recursos computacionais. Porém, ela acaba sendo menos precisa que outras funções de ativação. [51]

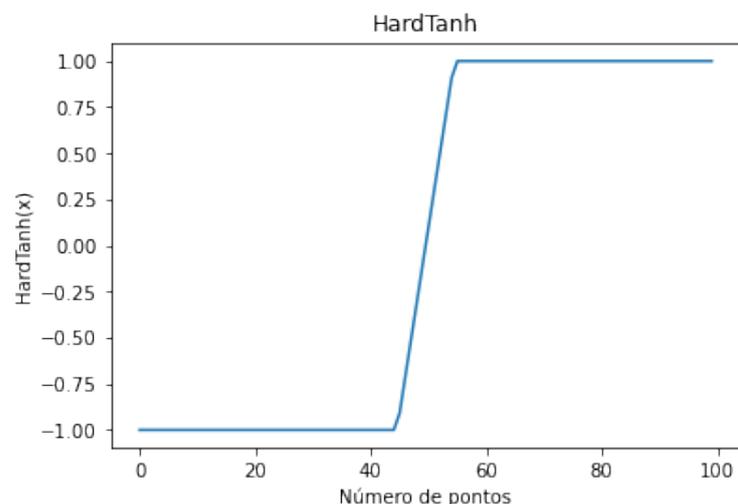


Figura 22: Visualização da função Hardtanh.

A Unidade Linear Exponencial, ou ELU, usa uma curva logarítmica para definir os valores negativo. Com isso, ajuda a rede a direcionar pesos e tendências na direção certa. [50] Uma desvantagem técnica das ELUs é que computar a função exponencial é uma operação ineficiente, tornando redes neurais com ELUs um pouco mais lentas. [52]

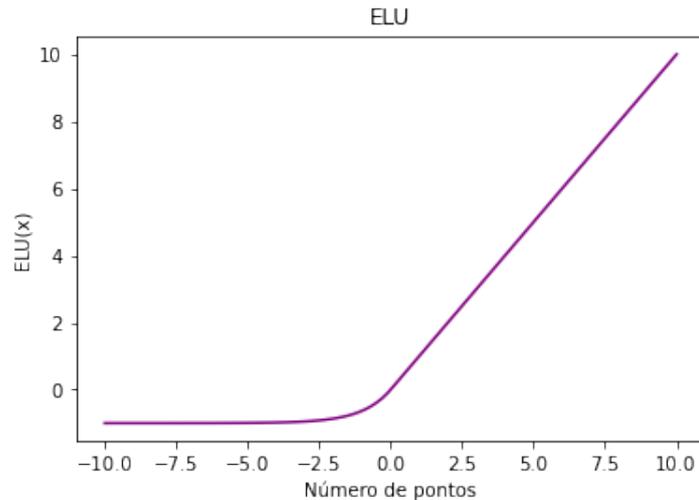


Figura 23: Visualização da função ELU.

- **Algoritmo de Otimização:**

Algoritmos de otimização, por sua vez, são ferramentas matemáticas que são executadas de forma iterativa, comparando várias soluções possíveis, até que uma solução ótima, ou pelo menos satisfatória, seja encontrada. São eles Gradiente Descendente Estocástico (SGD), Adam, RMSProp, entre outros.

Encontrar uma boa combinação de hiperparâmetros para uma rede neural é um processo trabalhoso, porém existem passos e orientações que facilitam essa atividade. Como por exemplo: utilizar uma arquitetura inicial simples e valores de hiperparâmetros padrão, depois ir diversificando e a deixando mais complexa conforme a necessidade e desempenho da rede a partir das métricas. [53]

g Visão Computacional

A visão computacional é uma tecnologia usada por máquinas para reconhecer imagens automaticamente e descrevê-las com precisão e eficiência. Hoje, os sistemas de computador têm acesso a grandes quantidades de dados de imagens e vídeos criados por smartphones, câmeras de trânsito, sistemas de segurança e outros dispositivos. Os aplicativos de visão computacional usam inteligência artificial e aprendizado de máquina (AI/ML) para processar com precisão esses dados para reconhecimento de objetos e reconhecimento facial, bem como classificação, recomendação, monitoramento e detecção. [54]

Embora a tecnologia para processamento de informações visuais já exista há algum tempo, grande parte do processo requer intervenção manual, que é demorada e sujeita a erros.

Por exemplo, a visão computacional facilita a análise de imagens capturadas por rovers ajudando na identificação de características geológicas dos planetas explorados. Isso exige milhares de imagens contendo pontos de dados importantes, como o tipo de superfície e rochas em diversos locais. Automatizar essas tarefas requer alto poder computacional porque os dados da imagem não são estruturados e a organização do computador é complexa. Como resultado, as aplicações de visão são caras e estão fora do alcance da maioria das organizações.

h Teoria de Controle Clássica

A teoria clássica de controle é uma área da teoria de controle na qual se trata do comportamento de sistemas dinâmicos. Dada uma entrada qualquer é analisado o comportamento do sistema estudado de acordo com seu feedback.

Um sistema clássico pode ser regido pela seguinte diagramação:

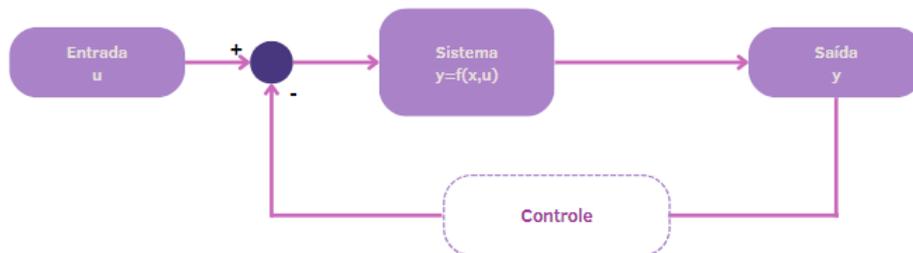


Figura 24: Diagrama de blocos de um sistema de Controle Genérico.

O objetivo usual da teoria de controle é controlar um sistema, muitas vezes chamado de planta, que possui um modelo matemático, de modo que sua saída siga um sinal de controle desejado, chamado de referência, que pode ser um valor fixo ou variável. Para isso é projetado um controlador que monitora a saída e a compara com a referência. A diferença entre a saída real e a desejada, chamada de sinal de erro, é aplicada como feedback à entrada do sistema, para aproximar a saída real da referência. [55]

A teoria clássica de controle lida com sistemas lineares invariantes no tempo (SLIT) de entrada e saída única (SISO) e geralmente trabalham no domínio da frequência ao invés do tempo, usando ferramentas como transformada de Laplace e Z. [55]

1 Planta

A planta do sistema de controle ou apenas sistema é definida como sendo a parte do sistema cujo comportamento quer ser controlado. Essa parte contém a dinâmica do sistema, ou seja, a descrição matemática de seu comportamento [56]. Em outras palavras, é o processo ou sistema real que está sujeito a influências externas e que precisa ser regulado ou controlado para alcançar um objetivo específico.

Um exemplo, em um sistema de controle de velocidade de um motor de uma aeronave, a "planta" seria o próprio sistema eletro-mecânico do motor. O objetivo pode ser controlar a velocidade dele para garantir um funcionamento suave e eficiente.

2 Tipo de Malha do Sistema

Para o atual estudo existem dois principais tipos de malhas para sistemas de controle: a malha aberta e a malha fechada. A malha aberta é um sistema de controle em que as ações de controle são realizadas com base em um plano pré-estabelecido, não levando em conta o feedback do processador, ou seja, leitura de um sensor. É como seguir um roteiro sem considerar o resultado real [57].

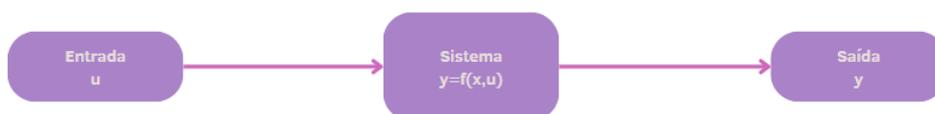


Figura 25: Exemplo de Malha Aberta de um sistema.

Já na malha fechada, a informação sobre a variável controlada é utilizada para ajustar as variáveis de entrada, permitindo uma resposta contínua às mudanças no processador [57].

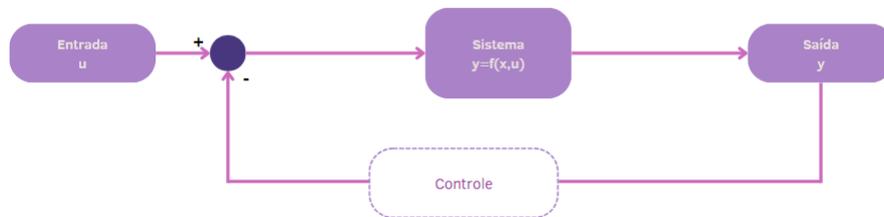


Figura 26: Exemplo de Malha Fechada de um sistema.

3 Controle PID

De acordo com a Caltech™: “O controlador PID é a forma mais comum de feedback. No controle de processos hoje, mais de 95% das malhas de controle são do tipo PID, a maioria das malhas são na verdade controle PI. O controle PID é frequentemente combinado com lógica, funções sequenciais, seletores e blocos de funções simples para construir complexos sistemas de automação usados para produção, transporte e fabricação de energia. Muitas estratégias de controle sofisticadas, como o controle preditivo de modelo, também são organizadas hierarquicamente. O controle PID é usado no nível mais baixo; o controlador multivariável fornece os pontos de ajuste aos controladores no nível inferior. O controlador PID pode, portanto, ser considerado o “pão com manteiga” da engenharia de controle. É um componente importante na caixa de ferramentas de todo engenheiro de control.” [58]

O control PID é dado pela seguinte equação:

$$u(t) = K[e(t) + \frac{1}{T_i} \int_0^t e(x) dx + T_d \frac{d(e(t))}{dt}] \quad (7)$$

Onde u é o sinal de controle e E é o erro de controle.

O sinal de controle é, portanto, uma soma de três termos: o termo P , que é proporcional ao erro E , o termo I , que é proporcional à integral do erro E , e o termo D que é proporcional à derivada do erro E . Os parâmetros do controlador são ganho proporcional K , tempo integral T_i e tempo derivativo T_d . [58]

A parte derivada, está associada ao pico da função durante o tempo. Ela serve para aumentar ou reduzir o tempo de subida da função. A parte proporcional, funciona apenas como um ganho, ela pode implicar na ênfase ou redução da resposta ao impulso de entrada. A parte integral, por fim, está relacionada a redução do erro em regime permanente. Note que, dependendo da configuração essas explicações podem mudar.

i Controle Extremal

A busca extremal foi uma ferramenta popular em aplicações de controle nas décadas de 1940 e 1960 e que vem ganhando espaço hoje em dia. Diferente de um controle adaptativo clássico esse modelo não é baseado em um modelo. Com isso, fornece uma alternativa rigorosa e de alto desempenho aos métodos de controle que envolvem redes neurais. [59]

Em controle extremal o conceito parte do objetivo de fazer com que um sistema tenda a operar em uma condição extrema, como maximizar ou minimizar uma determinada métrica de desempenho, podendo ser RMSE ou acurácia no caso desse atual estudo. Esta abordagem é frequentemente usada em situações onde a otimização do comportamento de um sistema é desejada e é desejado manter o sistema nessas condições operacionais extremas.

De acordo com Miroslav Krstić, teórico em controle e ilustre professor de Engenharia Mecânica e Aeroespacial na Universidade da Califórnia: “A busca de um extremo é aplicável em situações onde existe uma não linearidade no problema de controle, e a não linearidade tem um mínimo ou máximo local. A não linearidade pode estar na planta, como uma não linearidade física, possivelmente manifestando-se através de um mapa de equilíbrio, ou pode estar no objetivo de controle, adicionado ao sistema através de uma função de custo de um problema de otimização. Consequentemente, pode-se usar a busca de

extremo tanto para ajustar um ponto de ajuste para atingir um valor ideal de saída, quanto para ajustar parâmetros de uma lei de feedback". [59]

Por exemplo, dada uma entrada u de um sistema genérico que gera uma saída qualquer. Com feedback negativo, contrói-se um sistema de controle que leve o sistema a se comportar como o usuário deseja. Se o usuário quer tomar uma abordagem satisfatória para resolver o problema, ele irá definir uma função de custo, mínimo global, que ele busca minimizar, ou uma função objetiva para maximizar, máximo global. Após isso, ele irá definir os parâmetros das funções que realizem esta tarefa. [60]

No geral, o controle extremal consiste em conduzir ativamente um sistema em direção a condições operacionais ideais ou desejadas, muitas vezes em ambientes dinâmicos e incertos. [61]

Exemplificando um pouco mais. Dado um sistema de pouso de uma nave espacial em um exoplaneta onde a atmosfera não é turbulenta. A entrada do sistema é a intensidade dos propulsores da nave e a saída é a desaceleração para o pouso, o controle seria os parâmetros necessários e ideais para estabilizar os propulsores a fim de desacelerar o sistema da nave sem que ela oscile e quebre. Todavia, é necessário maximizar essa desaceleração a fim de gastar menos combustível possível na manobra de pouso para ter o suficiente para a volta. Essa maximização, é a função de custo. Com ela é possível obter o melhor dos dois mundos: gastar menos combustível e ter um pouso suave.

Note que nessa exemplificação o estudo de caso não é linear, o desgaste de combustível depende da dinâmica do sistema e parâmetros externos que merecem estudos a parte. Logo apenas uma solução fixa para esse problema não é viável visto que uma modelagem matemática é bastante complexa.

No controle extremal, o algoritmo de controle perturba continuamente o sistema e observa as mudanças resultantes na métrica de desempenho em sistemas dinâmicos. Ao analisar essas mudanças, o algoritmo pode determinar a direção na qual a métrica de desempenho melhora e ajustar as entradas de controle de acordo.

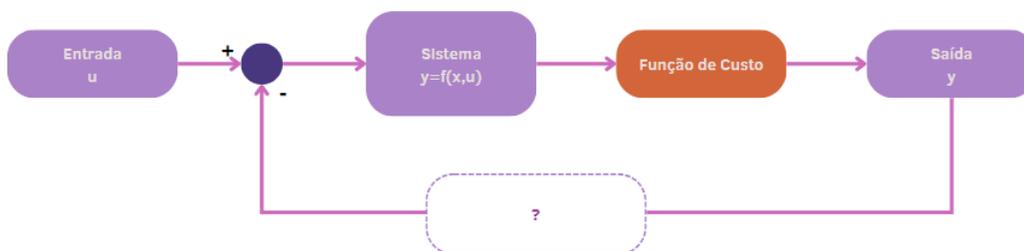


Figura 27: Diagrama de blocos de um sistema final de controle extremal.

Onde "?" é o sistema de controle/desmodulação e atualização de parâmetros do sistema, como será explicado abaixo.

Este processo iterativo permite que esse método de controle se adapte às mudanças nas condições operacionais e aos distúrbios, tornando-o particularmente útil para sistemas com dinâmica desconhecida ou parâmetros variantes no tempo. Ao buscar continuamente o extremo da função de custo, esta estratégia de controle pode ajudar a otimizar o desempenho do sistema em tempo real. [61]

Este processo iterativo nada mais é que tomar uma suposição inicial de uma variável do sistema e ir aumentando-a até achar um ponto de máximo e mínimo.

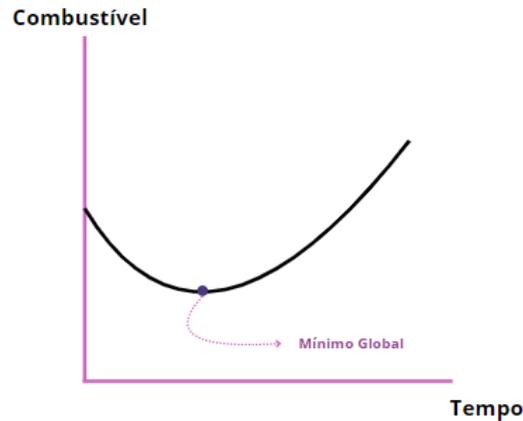


Figura 28: Curva genérica de consumo de combustível ao longo do tempo.

Em termos matemáticos, achar o ponto onde a derivada é zero. Mais especificamente, para garantir o ponto global, garantir que ambas primeira e segunda derivadas são iguais a zero. Ou seja:

$$\frac{df}{dx} = 0 \quad (8)$$

$$\frac{d^2f}{dx^2} = 0 \quad (9)$$

No geral, o controle extremal é uma ferramenta poderosa para otimizar o desempenho do sistema sem exigir um modelo detalhado do sistema, tornando-o uma abordagem valiosa para uma ampla gama de aplicações de controle.

Para esse estudo, a fim de modelar um algoritmo com controle extremal, é necessário entender os estágios para ajustar seus parâmetros definidos pelo MATLAB™: [9]

- **Modulação:** Perturbe o valor do parâmetro que está sendo otimizado usando um sinal senoidal de baixa amplitude.
- **Resposta do Sistema:** O sistema que está sendo otimizado reage às perturbações dos dos parâmetros. Esta reação causa uma mudança correspondente no valor da função objetivo.
- **Demodulação:** Multiplique o sinal da função custo por uma senóide com a mesma frequência do sinal de modulação. Este estágio inclui um filtro passa-alta opcional para remover polarização do sinal da função objetivo.
- **Atualização de parâmetro:** Atualize o valor do parâmetro integrando o sinal demodulado. O valor do parâmetro corresponde ao estado do integrador. Este estágio inclui um filtro passa-baixa opcional para remover ruído de alta frequência do sinal demodulado.

Com isso, temos a seguinte modulação dada pelo Simulink™:

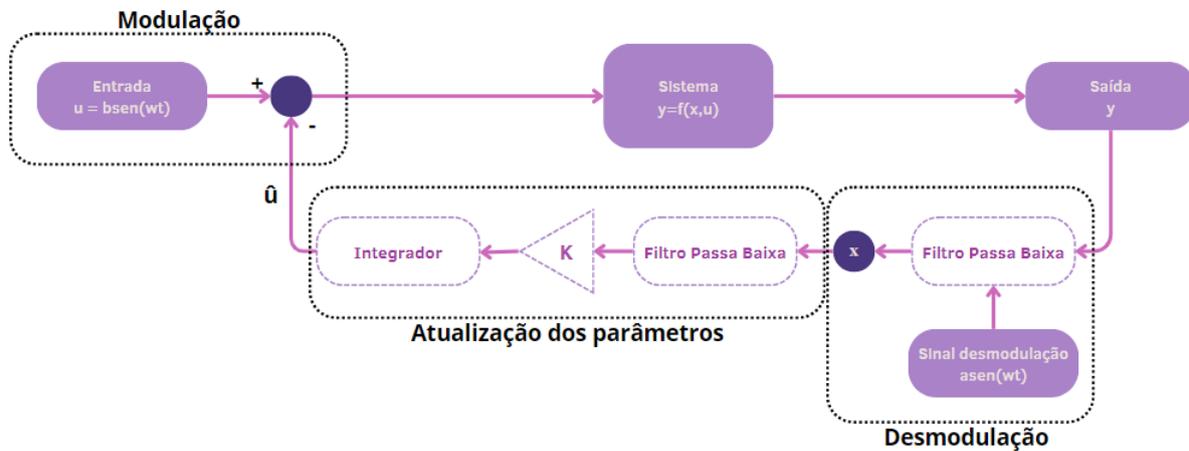


Figura 29: Diagrama em blocos do controle extremal [9]

Onde: [9]

- \hat{u} é o valor estimado do parâmetro.
- u é o sinal de modulação
- $y = f()$ é a saída da função que está sendo maximizada, ou seja, a função objetivo.
- ω é a frequência de força dos sinais de modulação e demodulação.
- $b \cdot \sin(\omega t)$ é o sinal de modulação.
- $a \cdot \sin(\omega t)$ é o sinal de desmodulação.
- k é a taxa de aprendizagem.

Porém, existem problemáticas. O primeiro ponto de observação é que ao inicializar o controle é necessário procurar o controle global e não local. Se uma função possui múltiplos máximos locais o sistema modelado não consegue distinguir pode não conseguir definir se aquele ponto é o ponto mais baixo/alto da função. O segundo problema é achar os parâmetros ideais para fazer o sistema convergir rapidamente, dependendo do sistema são muitos parâmetros a serem testados. [60] O último ponto a comentar nesse estudo é achar a função de custo ideal para o sistema, dependendo da função escolhida, a resposta será diferente e talvez não dê o resultado esperado.

j Função de Custo

Como dito na seção anterior, a função custo representa o objetivo ou meta que o sistema de controle pretende otimizar. Em termos mais específicos, a função custo normalmente representa o critério de desempenho que o sistema de controle procura otimizar. Isto poderia ser minimizar o consumo de combustível, minimizar o erro do sistema de atuação ou alcançar algum outro resultado desejado. Uma abordagem comum é o uso de funções de custo que integram o erro ao longo do tempo. Este tipo é frequentemente usada em sistemas de controle de feedback para penalizar desvios do ponto de referência.

Para o atual estudo foram analisadas as seguintes funções de custo:

1 ISE

Essa função nada mais é que a integral do erro quadrático. Ela está associada ao Controle Integral, ou seja, nos controladores PI, ou seja, essa parte tende a diminuir o error em regime permanente de um sistema. A integral da função de custo do erro quadrático é definida como: [62]

$$J = \int_0^t e(t)^2 dt \quad (10)$$

Onde:

- J é a função de custo.
- $e(t)$ é o erro no tempo
- t , definido como a diferença entre o setpoint desejado (ou trajetória de referência) e a saída real do sistema.

Esta função de custo representa o erro quadrático acumulado ao longo do tempo. Minimizar esta função de custo significa minimizar o erro acumulativo, o que muitas vezes leva a um melhor desempenho de rastreamento e à precisão do estado em regime permanente do sistema.

2 IAE

Chamada de Integral de Erro Absoluto (IAE), esta função de custo é frequentemente utilizada nos casos em que o erro absoluto é considerado mais apropriado do que o erro quadrático [63]. A função de custo Integral de Erro Absoluto (IAE) é definida como:

$$J = \int_0^t |e(t)| dt \quad (11)$$

Onde:

- J é a função de custo.
- $e(t)$ é o erro no tempo t , definido como a diferença absoluta entre o ponto desejado e a saída real do sistema.

Esta função de custo representa o erro absoluto acumulado ao longo do tempo. Minimizar esta função de custo significa minimizar este erro, o que pode ser vantajoso em certas aplicações de controle onde grandes erros são indesejáveis.

3 ITAE

ITAE significa Integral do Tempo multiplicado pelo Erro Absoluto. É outra variação da função de custo e está relacionada a sintonia de controladores PID. A função de custo ITAE é frequentemente preferida quando se deseja um equilíbrio entre resposta rápida e excesso mínimo [63]. Contudo, é utilizada apenas para sinais amostrados.

A função de custo Integral de Tempo multiplicado por Erro Absoluto (ITAE) é definida como:

$$J = \int_0^t t * |e(t)| dt \quad (12)$$

Onde:

- J é a função de custo.
- $e(t)$ é o erro no tempo t , definido como a diferença absoluta entre o ponto desejado e a saída real do sistema.

Esta função de custo representa o produto do tempo e o erro absoluto ao longo do tempo, essencialmente dando mais peso aos erros que ocorrem no início do processo.

Ao minimizar a integral do tempo multiplicada pelo erro absoluto ao longo do tempo, o sistema de controle visa obter uma resposta rápida enquanto mantém o overshoot mínimo.

k Ferramentas de Avaliação de Sistemas

A fim de analisar a dinâmica do sistema como o comportamento de seus pólos e zeros, foram utilizadas algumas ferramentas de visualização como as mostradas a seguir.

1 Retrato de Fase

Um retrato de fase, em essência, é uma representação visual de como um sistema evolui ao longo do tempo. É particularmente útil na compreensão de sistemas descritos por equações diferenciais. [64]

Um *Phase Portrait Plotter* é uma ferramenta usada para gerar retratos de fase de sistemas dinâmicos. Ele permite visualizar o comportamento de um sistema plotando suas variáveis de estado umas contra as outras no espaço de fase. Os retratos de fase são especialmente úteis para analisar estabilidade, pois podem revelar se um sistema se estabiliza num determinado estado (equilíbrio) ou se continua oscilando para sempre. Também são usados para mostrar padrões repetitivos (como o balanço de um pêndulo). Além disso, é possível analisar a sensibilidade às condições iniciais ou seja, pequenas mudanças no estado inicial podem levar a trajetórias muito diferentes em alguns sistemas. [64]

Dentro do Retrato de Fase existe o conceito de pontos singulares, que nada mais são do que pontos de equilíbrio do sistema, ou seja, o ponto em que o sistema pode se manter para sempre. Para sistemas lineares há apenas uma solução disponível, já sistemas não lineares, o caso de estudo, pode haver mais de um ponto. [65]. Para sistemas lineares a estabilidade do sistema pode ser descrita pela natureza dos pontos de singularidade.

Para sistemas não lineares, devido sua complexidade, sistemas podem ser analisados também olhando os ciclos-limites dos mapas. O ciclo-limite dos sistemas é definido como uma curva isolada e fechada. A natureza dessa curva é fechada indicando um movimento cíclico. [65]

Dependendo do sistema, o limite de ciclos pode ser classificado como:

- Limite de Ciclo Estável: Todas trajetórias no limite do ciclo convergem quando $t \rightarrow \infty$
- Limite de Ciclo Instável: Todas trajetórias ao redor do limite do ciclo divergem $t \rightarrow \infty$
- Limite de Ciclo Semi-Estável: Algumas trajetórias ao redor convergem, enquanto outras divergem em $t \rightarrow \infty$

Deste modo é possível avaliar as diversas regiões de convergência/divergência de um sistema não linear, garantindo uma análise local/global, ou seja, sabendo que uma dada região de ciclo é convergente, pode-se limitar as entradas e saídas do sistema à essa região, garantindo estabilidade local, ou ainda, avaliando-se que não há região de divergência, pode-se garantir que o sistema é globalmente convergente. Esta teoria ainda é muito discutida, aliada à Funções de Lyapunov para prova de estabilidade global, por exemplo.

2 Diagrama de Bode

Um gráfico de Bode é um gráfico da resposta de frequência de um sistema que consiste em dois gráficos: um mostrando a resposta de magnitude (em decibels e outro mostrando a resposta de fase (em graus) em função da frequência.

Os gráficos de Bode são utilizados para analisar o comportamento de sistemas. Eles fornecem uma visão sobre como um sistema responde a diferentes frequências, com isso é possível projetar e ajustar sistemas para atender às especificações de desempenho desejadas. [66]

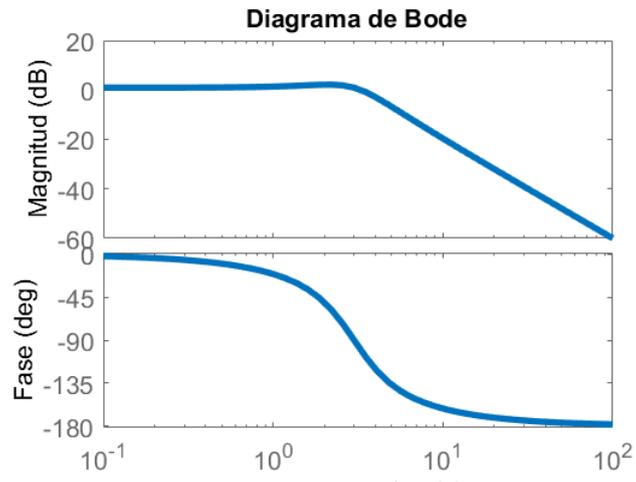


Figura 30: Exemplo de Diagrama de Bode. [10]

5 DISCUSSÃO: “A ARTE”

Nesta seção estão apresentados os tópicos de interesse para a criação dos modelos preditivos. Para tal desenvolvimento, inicia-se com a elaboração das Redes Neurais.

a Base da Dados: Rede Neural

Para o desenvolvimento das Redes Neurais foi utilizado a biblioteca PyTorch™ do Python™, além da SKLearn™, para avaliação de métricas e Seaborn para visualização de dados. O banco de dados foi retirado pela plataforma Kaggle™.

Adentrando um pouco mais na base de dados: Este conjunto de dados é um registro cumulativo de todos os “objetos de interesse” observados pelo Kepler - basicamente, todos os aproximadamente 10.000 candidatos a exoplanetas sobre os quais o Kepler fez observações. Este conjunto também possui um extenso dicionário de dados, fornecido pela NASA™.

O banco de dados possui um conjunto de colunas destacáveis que a NASA™ informou [67]:

- *kepoi_name*: Um KOI é um alvo identificado pelo Projeto Kepler que exibe pelo menos uma sequência semelhante a um trânsito na fotometria de série temporal do Kepler que parece ser de origem astrofísica e inicialmente consistente com uma hipótese de trânsito planetário
- *kepler_name*: Têm como objetivo indicar claramente uma classe de objetos que foram confirmados ou validados como planetas – um avanço em relação à designação de candidato a planeta.
- *koi_adisposition*: A disposição da literatura em relação a este candidato a exoplaneta.
- *koi_pdisposition*: A disposição que a análise de dados do Kepler tem em relação a este candidato a exoplaneta.
- *koi_score*: Um valor entre 0 e 1 que indica a confiança na disposição KOI. Para CANDIDATOS, um valor mais alto indica mais confiança na sua disposição, enquanto para FALSOS POSITIVOS, um valor mais alto indica menos confiança nessa disposição.

O *koi_score* será utilizado para o desenvolvimento da Rede Neuro-Fuzzy, porém no momento não será utilizado.

Pelo Kaggle, a melhor previsão obteve 95% de acurácia, o objetivo desse estudo é melhorar esse valor.

A limpeza dos dados foi baseada no trabalho de Ismael Araujo. Nele, ele capta as 9.564 observações e 49 *features* e os deixa de mais fácil entendimento [36].

Nesse trabalho, ele renomeia algumas colunas do dataset para nomes de mais fácil compreensão visto que muitos possuem nomes técnicos. Desse jeito, analisando a base de dados, é compreendido a significância de cada uma das informações presentes. A NASA™ dispôs o *dataset* com todas as informações de cada categoria, logo é possível entender quais informações serão usadas para o escolha dos alvos de predição. Colunas não necessárias foram dispostas sem que a integridade da quantidade de dados fosse prejudicada. Linhas nulas também foram descartadas [36].

No final do processo de limpeza, foram disponibilizadas 7.803 amostras com 38 *features*, uma delas é *ExoplanetCandidate* que se tornará a variável principal de estudo. Ela fornece a informação se a amostra é candidato ou confirmado [36].

b Rede Neural

Com os dados limpos, foi separado a porcentagem de teste e treinamento. Essa porcentagem foi definida a partir de várias tentativas, primeiro foi testado com 40% de treinamento e 60% de teste que foram o que obtiveram o melhor resultado. Ao testar com variações de 30-70 e 70-30, o desempenho da rede em questão de acurácia, caía mas não drasticamente. A fim, de manter tal performance e melhorar com outros hiperparâmetros o resto dos testes foram realizados com 40-60.

Além disso, foram usadas seis redes neurais diferentes: um Perceptron mais simples e nove Perceptrons Multicamadas, um com duas camadas ocultas e o resto com três porém com funções de ativação diferentes. Vale ressaltar, que foram testadas funções de ativações diferentes na rede de duas camadas, mas a performance se manteve. No ensaio foram escolhidas funções como: Tanh, Sigmoid, Softmax, ReLU,

LogSoftMax, HardTanh e ELU. Foi usada também HardShrink a fim de curiosidade e a acurácia foi de 0.58.

Todas as Redes Neurais utilizadas possuíam em sua camada de saída uma Função Sigmóide com um limite de 0.5 para a classificação final.

O treinamento foi feito utilizando o "Adam" como otimizador e utilizando a Raíze do Erro Quadrático Médio (RMSE) como função de perda. Foram feitos dois treinamentos diferentes, um com 60 épocas e outro com 50 épocas. Todas as outras configurações do treinamento permaneceram as mesmas.

Dessa forma foram feitos mais de 46 experimentos para gerar os resultados finais.

c Base da Dados: Rede Neuro-Fuzzy

Para o desenvolvimento da Rede Neuro-Fuzzy foi utilizada tanto a biblioteca da Rede Neural, quanto a biblioteca de Python™ voltada a astronomia, Astropy™ e também uma nova base de dados referentes a da Rede Neural. Essa escolha foi feita para comparar quais formatos de informações gerariam uma melhor análise. Da biblioteca da Rede Neural foram utilizadas as colunas com informações necessárias para determinação de um exoplaneta, tais como: temperatura de equilíbrio, raio do planeta, informações de trânsito e a disposição da literatura em relação a este candidato a exoplaneta. Com esses dados e a informação se confirmação de exoplaneta é possível gerar um sistema neuro-fuzzy capaz de classificar exoplanetas.

A biblioteca Astropy™, fornece dados mais nichados para astronomia do que o da Rede Neural. Nele foi utilizado a ferramenta de periodograma "box mínimos quadrados" (BLS), que é uma ferramenta estatística usada para detectar exoplanetas em trânsito. O método BLS encontra candidatos a trânsito modelando um trânsito com quatro parâmetros: período, duração, profundidade e um tempo de referência. Nesta implementação, o tempo de referência é escolhido para ser o tempo intermediário do primeiro trânsito na linha de base observacional [68]. Esses parâmetros são mostrados no esboço a seguir:

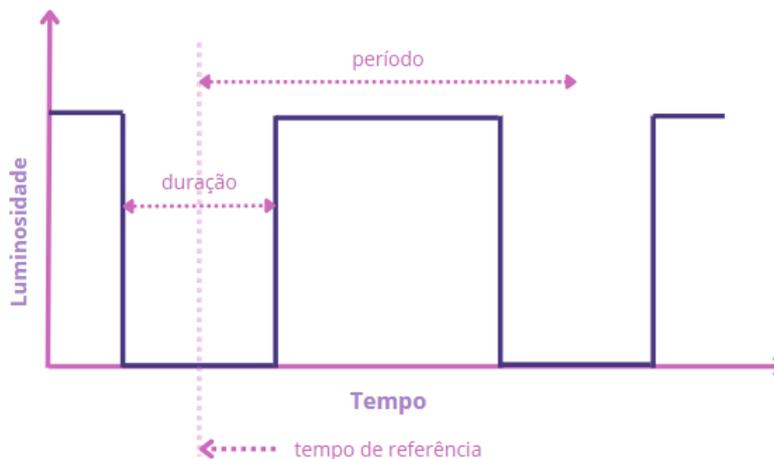


Figura 31: Formato da análise de parâmetros do BLS

Assumindo que as incertezas no fluxo medido são conhecidas, independentes e gaussianas, o fluxo em trânsito (y_{in}) pode ser calculado como:

$$y_{in} = \frac{\sum_{in} y_n / x_n^2}{\sum_{in} 1 / x_n^2} \quad (13)$$

onde y_n são as medidas de brilho, x_n são as incertezas associadas e ambas as somas são calculadas sobre os pontos de dados em trânsito. [68]

Da mesma forma, o fluxo fora de trânsito de máxima probabilidade (y_{out}) é dado por:

$$y_{out} = \frac{\sum_{out} y_n/x_n^2}{\sum_{out} 1/x_n^2} \quad (14)$$

Usando esses resultados, o log de probabilidade de um modelo de trânsito em um determinado período P , duração T e tempo de referência t_0 é: [68]

$$\log \mathcal{L}(P, T, t_0) = -\frac{1}{2} \sum_{in} \frac{(y_n + y_{in})^2}{x_n^2} - \frac{1}{2} \sum_{out} \frac{(y_n + y_{out})^2}{x_n^2} + Constante \quad (15)$$

Onde C é uma constante de integração discreta. [69] [70] Nos bastidores, esta implementação minimiza o número de cálculos necessários, pré-classificando as observações em uma grade fina seguindo [68] e [11].

Com essas ferramentas, conseguimos extrair e solidificar nossa base de dados com as informações necessárias para encontrar um planeta em trânsito. Note que, as análises são feitas a partir de determinado exoplaneta escolhido pelo usuário, não a base de dados inteira do Kepler. Escolhendo o K2-3, obtivemos a primeira análise de trânsito.

Os dados necessários para análise foram principalmente a força e período. Porém, a fim de gerar mais regras fuzzy, foram necessárias manipulações na base. Tais manipulações serão explicadas na subseção a seguir.

Para nova biblioteca derivada da Rede Neural foi usado o arquivo de Mikulski™ [71] derivado do Kaggle™. O autor criou conjuntos de dados das estrelas analisadas, os limpou e processou transpondo a coluna $PDC_{SAPFLUX}$, depois mesclou no *dataframe*. No momento da limpeza, o autor notou que era improvável que alguns conjuntos contivesse quaisquer exoplanetas-estrelas não descobertos. Portanto, todas as estrelas (ou seja, todas as fileiras) que não foram confirmadas como hospedeiras de um exoplaneta foram rotuladas com 0. Isso representou mais de 99% das observações. No total, há 42 observações rotuladas com 1.5 no conjunto de teste e 37 no conjunto de treinamento.

d Rede Neuro-Fuzzy

Para a criação da Rede Neuro-Fuzzy, foi utilizada a biblioteca ANFIS do MATLAB™. Nela aplica-se a inferência fuzzy a um sistema no qual já possui uma coleção de dados de entrada/saída contudo, não existe necessariamente uma estrutura de modelo predeterminada com base nas características das variáveis do sistema. Nesses casos, em vez de escolher arbitrariamente os parâmetros associados a uma determinada função de associação, o MATLAB™ permite ajustar os sistemas de inferência fuzzy usando técnicas de aprendizagem neuro-adaptativas semelhantes às usadas para treinar redes neurais [72].

A escolha da biblioteca do MATLAB™, se deu pelo fato de que algumas bibliotecas em Python™ para o Jupyter™, estavam gerando erros de importação de outras dependências internas ao rodar funções. Foram testados *anfis*, *keras*, *tensorflow* e *neurofuzzy* e não foi descoberto soluções para tais problemas. Além disso, o MATLAB fornece exemplos para guiar no uso de sua plataforma. Os utilizados para esse projeto foram: Previsão de consumo de combustível [73], Modelo de deslocamento suburbano usando agrupamento subtrativo e ANFIS [8] e Cancelamento adaptativo de ruído usando ANFIS [74].

Durante as tentativas em Python™, foi selecionada a base da Rede Neural e foram definidas duas variáveis de entrada e uma de saída fuzzy. Foram num total sete regras fuzzy definidas e as divisões de treinamento e teste da base foram mantidas. Contudo, devido as condições citadas a cima, não foi possível completar essa rede, por isso se seguiu com MATLAB™.

Para o banco de dados baseado na Rede Neural, como dito anteriormente, foram determinadas variáveis de entrada e saída, também variáveis de validação de entrada/saída do sistema. Essas validações foram extraídas da base original de forma a ter metade das informações de forma aleatória. Depois foi definido um modelo de *Clustering* Subtrativo. Ele é um algoritmo rápido para estimar o número de agrupamento em um conjunto de dados [8].

Para o exemplo foi usado um intervalo de influência de cluster de 0.3. Este valor indica o intervalo de influência de um agrupamento quando você considera o espaço de dados como um hiper-cubo unitário. Especificar um pequeno raio geralmente gera muitos pequenos agrupamentos nos dados, o que produz um FIS com muitas regras.

Uma vantagem importante de usar um método de agrupamento para encontrar regras é que as regras resultantes são mais adaptadas aos dados de entrada do que em um FIS gerado sem agrupamento. Essa adaptação reduz o número total de regras quando os dados de entrada têm uma dimensão elevada [8].

Em seguida foi determinados os hiperparâmetros da rede que melhor encaixavam no modelo. Foi utilizado a função de busca exaustiva para realizar uma pesquisa exaustiva dentro dos inputs disponíveis para selecionar o conjunto de inputs que mais influenciam no sistema. Essa função constrói um modelo ANFIS para cada combinação, treina-o para uma época e relata o desempenho alcançado [73].

No total foram realizados 9 experimentos variando os hiperparâmetros com essa base.

Para a base do Astropy™, foi utilizado uma outra abordagem. Com os dados extraídos da biblioteca, foi possível observar:

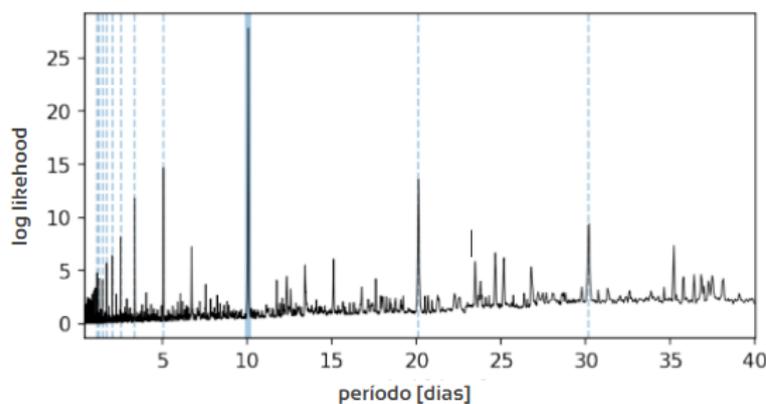


Figura 32: Periodograma do K2-3 [11].

A estrutura da Figura pode ver neste periodograma é bastante típica de sistemas com planetas em trânsito. Ele se relaciona por meio de $\log_{likelihood}$ que é o logaritmo da função de verossimilhança do modelo avaliado no ponto máximo (em relação ao espaço de parâmetros) e o período dos trânsitos. O período de pico é destacado com uma linha azul grossa e os harmônicos inteiros deste período são indicados com linhas azuis tracejadas [75]. Com essas informações em uma base de dados, foi necessário criar co-relações entre parâmetros a fim de gerar a informação e saída da rede neuro-fuzzy. Os períodos de pico são considerados grandes candidatos a exoplanetas, com isso para esse intervalo foi criada uma distribuição randomica de valores de 10.0 a 0.05, onde se tinham condicionais para obter maiores valores para picos e menores para observações próximas a zero, criando assim uma coluna parecida ao Disposition Score da outra base. Além disso, selecionou-se das variáveis de treinamento metade dos valores para teste de forma aleatória.

Foram feitos 21 testes envolvendo a base, eliminando e adicionando colunas para achar parâmetros que gerassem melhor resultado.

Para a base de Mikulski™ foi utilizado o projeto do $W\Delta$ [76]. Nela foram apenas selecionados as variável de saída da base original e diminui as variáveis de entrada, pois a base em seu formato bruto era muito pesada para o processamento do computador e as colunas não afetavam o desempenho da rede. Escolheu-se 12 colunas tanto para treinamento quanto para validação.

e Base de dados Controle Extremal

Para o desenvolvimento dessa parte foi utilizada a base de dados de imagem do telescópio Kepler provida pelo LightKurve™.

LightKurve™ extrai dados de séries temporais de brilho, curvas de luz, das pequenas imagens de estrelas coletadas pelos telescópios de planetas Kepler e TESS da NASA.

Por exemplo, para o atual estudo uma das amostras utilizadas foi a seguinte imagem:

Target ID: 200084891, Cadence: 128925

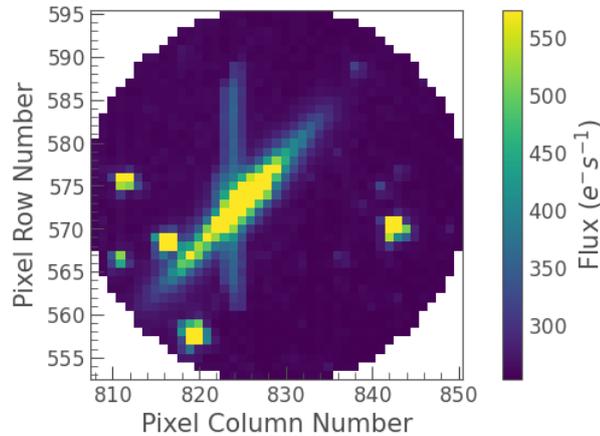


Figura 33: Arquivo em pixel da amostra da EPIC 200084891 extraído do LightKurve™.

Nessa figura já é possível ver que há uma coluna do que parece ser um “borrão” de saturação vertical no meio. Essa peculiaridade, na verdade, trata-se de crosstalk de uma estrela brilhante próxima. Esse é um caso de ruído de imagem que será trabalhado para construção dessa parte do estudo.

O objetivo dessa parte é transformar tais análises extraíndo seus ruídos e filtrando para deixar a imagem mais clara e não suscetível a erro de análise.

Foram usados também imagens não nítidas como a do ‘EPIC 211741417’ para garantir performance do modelo.

Target ID: 211741417, Cadence: 154331

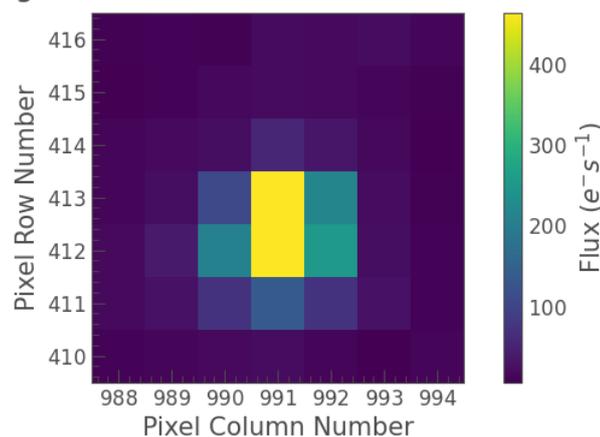


Figura 34: Arquivo em pixel da amostra da EPIC 211741417 extraído do LightKurve™.

f Controle Extremal

Durante o estudo de possíveis ruídos, foi analisado que o ruído de Poisson era um bom candidato, visto que ele realmente ocorre com o telescópio Kepler. A ideia de ruído aparece sempre que é considerado a precisão de uma medição. Geralmente pode-se melhorar a precisão da medição se é conhecido alguma informação sobre as estatísticas do ruído. [77]

Sabendo que a distribuição de Poisson pode ser dada pela equação abaixo: [77]

$$P(N) = e^{-r\tau} \frac{(r\tau)^N}{N!}. \quad (16)$$

Onde r é a taxa de chegada de fótons, τ é o tempo de observação e N a quantidade de fótons.

Para descobrir os parâmetros de Poisson foi utilizado a contagem média de fótons como parâmetro médio (λ), que caracteriza o ruído em cada pixel da imagem. A contagem de fótons média foi estimada considerando fatores como o brilho dos objetos observados (estrelas).

De acordo com o site da NASA™: "Embora estrelas mais brilhantes que $mV = 11,5$ saturem alguns pixels, K2 tem um bom desempenho em estrelas tão brilhantes quanto $mV = 4$, desde que o benefício científico justifique o grande número de pixels necessários para capturar o fluxo saturado que flui ao longo das colunas CCD." [78]

Com isso, foi assumido:

$$\lambda = r\tau \quad (17)$$

Se λ é aproximadamente 4 e supondo a situação mais simples onde τ é um segundo, temos r correspondente a 4. Note que, quanto maior $r * \tau$ mais Gaussiana será minha distribuição de Poisson, ou seja, ela será mais simétrica.

Com essa equação foi adotado o processo reverso para obter o estado ideal do meu sistema removendo o ruído.

Usando o MATLAB™ as imagens das observações de Kepler, foi extraído o ruído de Poisson delas. Com a matriz resultante de pixels foi gerada a equação aproximada de Poisson da imagens a fim de torná-la o sistema de referência do controle extremal.

Com a equação de ruído, foi utilizado a Transformação de Anscombe, aplicando-a à imagem com ruído de Poisson. Esta transformação estabiliza a variância do ruído de Poisson, o tornando uma aproximação de distribuição gaussiana. Ela é dada por:

$$f(x) = 2\sqrt{x + \frac{3}{8}} \quad (18)$$

Depois que a Transformação de Anscombe for aplicada, foi utilizado em MATLAB™ técnicas de eliminação de ruído para ruído gaussiano, usando filtros. Em termos matemáticos ela pode ser reduzida a:

$$G(x, y) = \frac{1}{2 * \pi * \tau} e^{-\frac{x^2 + y^2}{2\tau^2}} \quad (19)$$

Após a remoção de ruído, foi aplicado a transformação inversa de Anscombe para obter a imagem sem ruído. Ela é dada por:

$$f(x)^{-1} = \frac{x^2}{4} - \frac{3}{8} \quad (20)$$

Com isso obtivemos as duas equações responsáveis pela dinâmica real e ideal do sistema.

6 RESULTADOS: "A EXPOSIÇÃO"

Nesta seção serão discutidos e apresentados os resultados experimentais dos modelos desenvolvidos, junto com suas explicativas e gráficos.

a Rede Neural

No final do modelo, foi obtido a seguinte organização de hiperparâmetros: 40% dos dados foram para treinamento e 60% para teste com um *seed* de 6. Os recursos foram normalizados usando escalonamento min-max, a validação e o treinamento foram transformados para usar o *DataLoader* do PyTorch™.

Foram usadas cinco redes neurais diferentes para avaliar o resultado final: um Perceptron mais simples e quatro Perceptrons Multicamadas, um com duas camadas ocultas e o resto com três porém com funções de ativação diferentes: Tanh, Softmax, RELU e ELU.

Como dito acima, o treinamento foi feito utilizando o Adam como otimizador, 0.02 como taxa de aprendizado, 60 de número de épocas e utilizando o Erro Quadrático Médio (MSE) como função de perda. Todas as outras configurações do treinamento permaneceram as mesmas.

Com isso obtivemos:

Tabela 1: Lista de tipos de rede e resultados

Tipo de Rede	Acurácia	Recall	Precision	F1-Score
Single Layer Perceptron	0.9543	0.9348	0.9778	0.9558
Multi Layer Perceptron 2 Hidden Layers	0.9575	0.95	0.9691	0.9594
Multi Layer Perceptron 3 Hidden Layers (Tanh)	0.9615	0.9515	0.9752	0.9632
Multi Layer Perceptron 3 Hidden Layers (Softmax)	0.9599	0.9454	0.9780	0.9615
Multi Layer Perceptron 3 Hidden Layers (LogSoftmax)	0.9591	0.9515	0.9706	0.9609
Multi Layer Perceptron 3 Hidden Layers (HardTanh)	0.9567	0.95	0.9676	0.9587
Multi Layer Perceptron 3 Hidden Layers (ELU)	0.9599	0.9530	0.9707	0.9618

Também foi avaliado o sistema com base na sua Matriz de Confusão.

Tabela 2: Resultados das Matrizes de Confusão

Tipo de Rede	VP	FP	FN	VN
Single Layer Perceptron	574	14	43	617
Multi Layer Perceptron 2 Hidden Layers	568	20	33	627
Multi Layer Perceptron 3 Hidden Layers (Tanh)	572	16	32	628
Multi Layer Perceptron 3 Hidden Layers (Softmax)	574	14	36	624
Multi Layer Perceptron 3 Hidden Layers (LogSoftmax)	569	19	32	628
Multi Layer Perceptron 3 Hidden Layers (HardTanh)	567	21	33	627
Multi Layer Perceptron 3 Hidden Layers (ELU)	569	19	31	629

Note que VP significa Verdadeiros Positivos, que são os casos onde o modelo previu corretamente o que era de fato um exoplaneta, FP significa Falso Positivos, ou seja, casos em que o modelo previu incorretamente logo ele afirmou que era uma amostra de exoplaneta quando não era. Já, FN significa Falso Negativos, que são os casos em que o modelo previu incorretamente os casos em que não eram um planeta, quando na verdade era. E por último, VN significa Verdadeiros Negativos, ou seja, os casos em que o modelo previu corretamente o que não era um exoplaneta.

Com isso, olhando para desempenho relacionado a acurácia, é notável que com uma rede de três camadas com função de ativação teve melhor desempenho. Olhando para a matriz de confusão, vemos que essa mesma rede teve conseguiu desempenhar melhor a classificação de falsos positivos e negativos. Contudo, os resultados dos experimentos tiveram valores das métricas de desempenho muito próximos, concluindo que usando Redes Neurais menos complexas o resultado final será muito semelhante àquelas com mais camadas. Dessa forma, o uso de um Perceptron realizaria um bom algoritmo de análise para o problema que temos.

O gráfico de perda de treinamento e validação para os experimentos realizados acima:

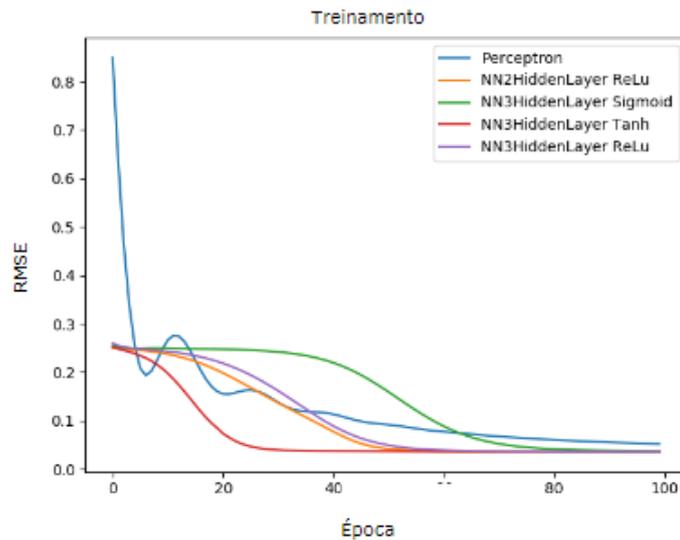


Figura 35: Função de perda durante treinamento.

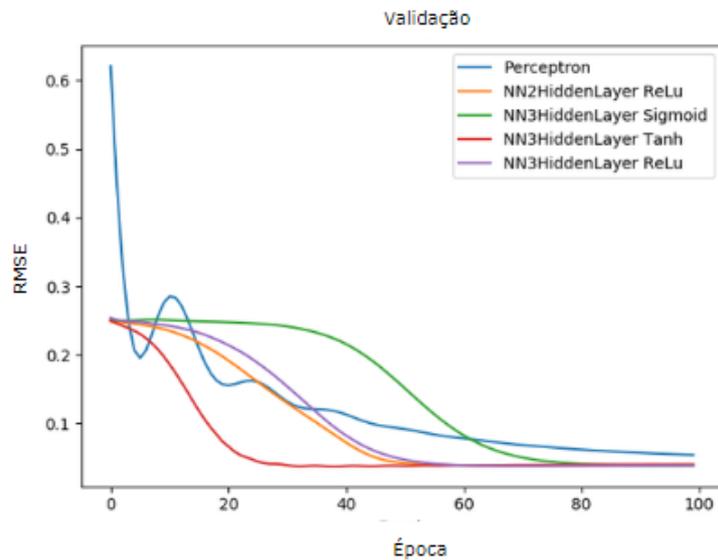


Figura 36: Função de perda durante validação.

É possível notar que para a rede de melhor desempenho Tanh, ele erra gradualmente com uma taxa decrescente até estabilizar com 0.1 em seu treinamento e menor que 0.1 em sua validação. Entretanto, todas as redes tendem a convergir ao erro em um ponto aproximados uns dos outros, confirmando a análise acima de que o uso de um Perceptron realizaria um bom algoritmo de análise para o problema que temos, mesmo que ele tenha errado mais no início.

b Rede Neuro-Fuzzy

No final, para a rede ANFIS, foram determinados hiperparâmetros diferentes para cada rede a fim de melhorar o desempenho.

Para a rede com a base Kaggle™ foram usados uma matriz (7803,7) de treinamento de entrada e outra

(7803,1) de saída. Para validação as bases foi dividida pela metade tornando-se (3902,7) e (3902,1) respectivamente. Foram testados os parâmetros de entrada do sistema por uma época com a função de busca exaustiva para determinar qual variável por si só pode prever melhor a saída. Contudo, esse tipo de busca não entregou os resultados esperados. Isso aconteceu visto que a rede com hiperparâmetros atuais erraram muito no início, logo o resultado não pode ser considerado. Dessa forma, se manteve as variáveis de entrada, e foram testados os resultados.

Para a rede com a base *Astrophy*TM foram usados uma matriz (63476,9) de treinamento de entrada e outra (63476,1) de saída. Para validação as bases foi dividida pela metade tornando-se (31738,9) e (31738,1) respectivamente. Nesta rede, também foram testados os parâmetros de entrada do sistema por uma época com a função de busca exaustiva para determinar qual variável por si só pode prever melhor a saída. Neste caso, a rede pode ser definida por apenas uma época.

Para a rede com a base $W\Delta$ foram usados uma matriz (5087,11) de treinamento de entrada e outra (5087,1) de saída. Para validação as bases foi determinado o mesmo número de linhas, mas selecionando colunas não utilizadas no treinamento, tornando-se (5087,11) e (5087,1) respectivamente. Todavia, os resultados da rede não foram promissores, por isso não serão expostos. Isso se deve ao fato da rede estar criando apenas uma regra para uma base tão extensa, logo está se tornando uma classificação binária. Para resolver essa situação, seria necessário expandir a variável de classificação de saída.

Foi aplicado à todas as bases três testes e três validações para medir o desempenho da rede. Note que os valores dos desempenhos 2 e 3, foram iguais, logo não havia necessidade de treinar e validar novamente a rede.

Tabela 3: Avaliação da rede ANFIS com a base KaggleTM por meio do RMSE

Tipo de Base	Treino 1	Validação 1	Treino 2 e 3	Validação 2 e 3
KAGGLE TM	0.3309	0.3299	0.3286	0.3280
Astrophy TM	0.0432	0.434	0.0244	0.0243

Tabela 4: Avaliação da rede ANFIS com a base KaggleTM por meio do MSE

Tipo de Base	Treino 1	Validação 1	Treino 2 e 3	Validação 2 e 3
KAGGLE TM	0.1095	0.1088	0.1080	0.1760
Astrophy TM	0.00187	0.1884	0.0005954	0.0005905

Tabela 5: Avaliação da rede ANFIS com a base KaggleTM por meio da Acurácia

Tipo de Base	Treino 1	Validação 1	Treino 2 e 3	Validação 2 e 3
KAGGLE TM	0.6991	0.700	0.6714	0.6720
Astrophy TM	0.9068	0.9438	0.9442	0.9666

Foi notados durante o processo, que o MATLABTM cria regras fuzzy em um formato otimizado, ou seja, uma regra possui um alto número de condições. Isso implicou no desempenho da rede, pois algumas das condições podiam gerar outras regras que melhorassem a resposta final do modelo. Para avaliar ainda mais a rede foi testado para todos o overfitting que é quando o erro de validação começa a aumentar enquanto o erro de treinamento continua a diminuir. Com isso, plotando os erros é possível notar que:

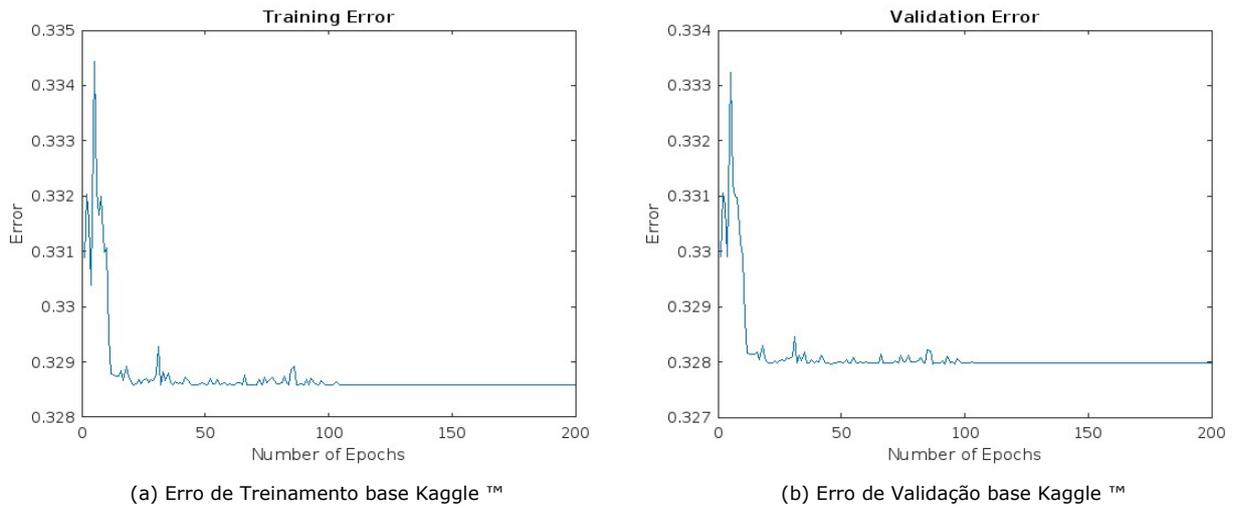


Figura 37: Simulação do erro da base Kaggle™

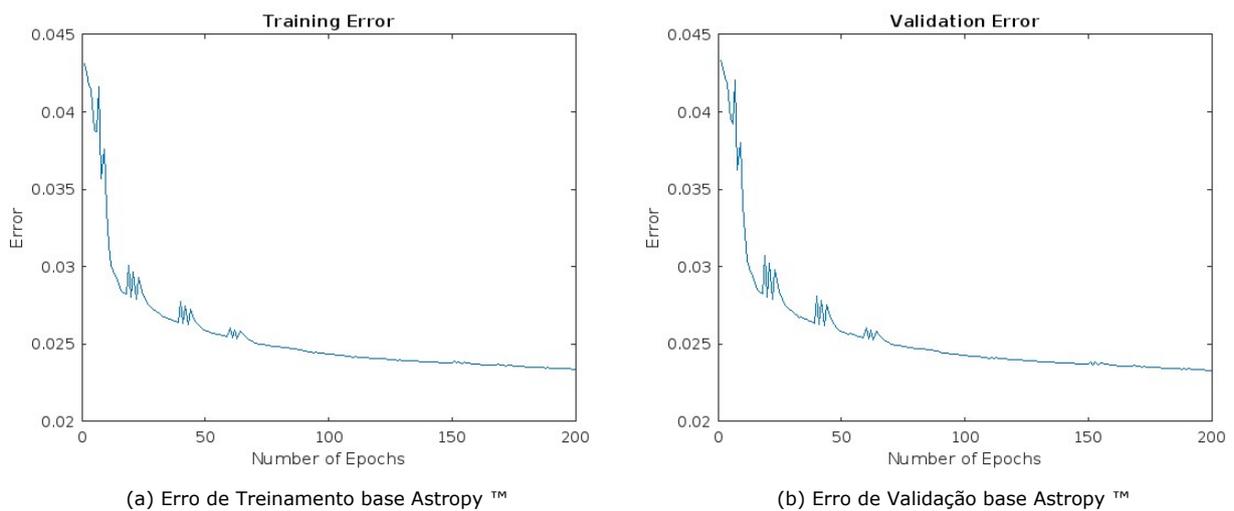


Figura 38: Simulação do erro base Astropy™

É possível notar que pela base Kaggle™ não teve *overfitting*, contudo até a época 200, demorou a convergir, errando muito no início. Já a Astropy™ foi por pouco que não teve *underfitting*.

c Controle Extremal

Com os dados das equações de distribuição de Poisson, foi modelado em blocos o sistema de controle extremal no Simulink™. O modelo atual, foi baseado no exemplo do próprio MATLAB™ *Extremum Seeking Control for Reference Model Tracking of Uncertain Systems*.

O diagrama em blocos projetado foi:

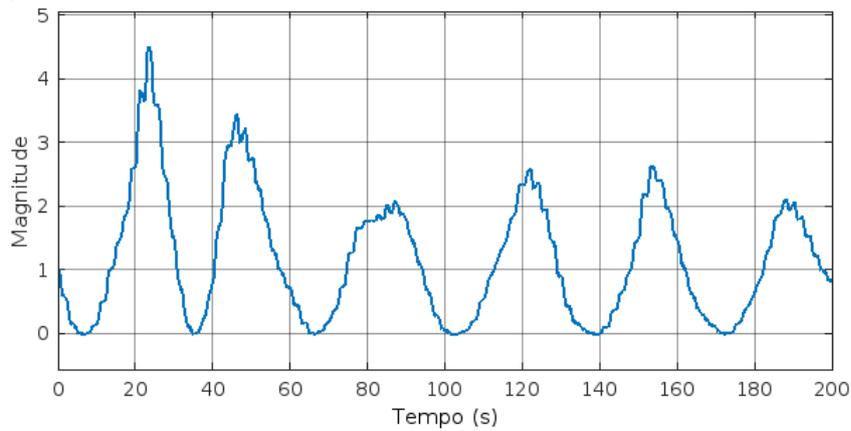


Figura 40: Função de Custo ISE com baixa taxa de aprendizado, $r\tau = 4$.

Com uma taxa de aprendizagem maior que 0.6 o sistema gerava um erro de configuração nas condições iniciais de setagem. Após algumas alterações de parâmetros, ao setar novamente essa mesma taxa de aprendizagem o resultado tendia a instabilidade.

Aumentando a frequência do sistema de controle extremal para acima de 500 rad/s fazia a função deixar de convergir, o sistema virava oscilatório. O mesmo comportamento era apresentado para frequências abaixo de 500 rad/s. A partir de 490 rad/s a função de custo começava a convergir, porém com alta oscilação até o final da simulação de 200s. Por isso, foi mantido $\omega = 500\text{rad/s}$

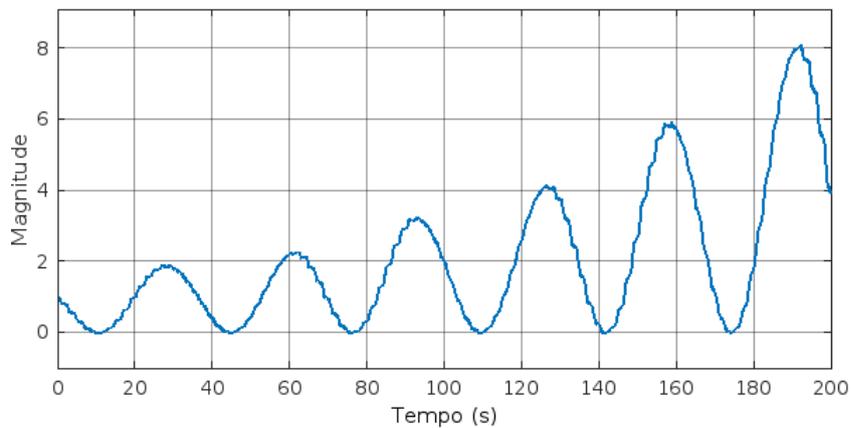


Figura 41: Função de Custo ISE com frequência de 600rad/s, $r\tau = 4$.

Com uma magnitude de demodulação 1, o sistema tende a regime permanente com baixa oscilação próxima a zero porém com alto esforço no início da simulação durante um período considerável de tempo, diminuindo esse valor, a oscilação diminui porém o alto esforço no início se mantém. Se o valor for abaixo de 0.4, o sistema tende a instável. Reduzindo para 0.5, o sistema demonstrou um resultado melhor, onde ainda há alto esforço, porém em um tempo reduzido.

Adicionando uma fase a essa demodulação instabilizava o sistema, por isso foi optado não mexer neste parâmetro.

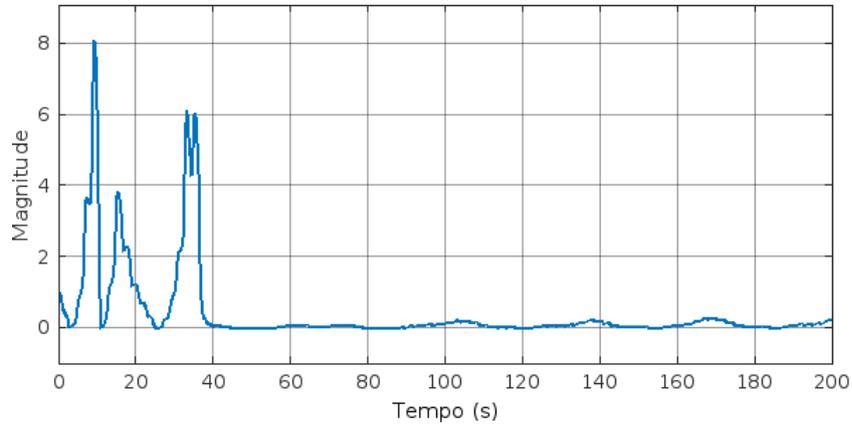


Figura 42: Função de Custo ISE com demodulação de 0.5, $r\tau = 4$.

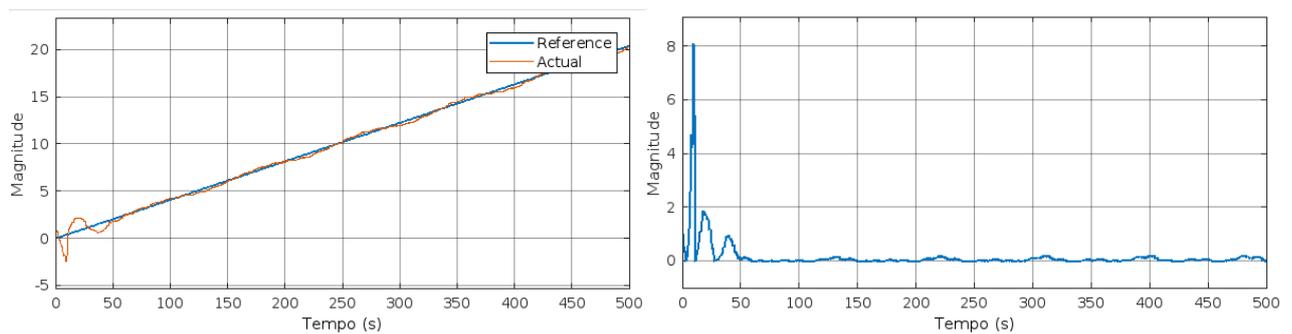
Aplicando um filtro passa-baixa e outro passa alta, foi possível manter o sistema em estabilidade por 500s. A fim de manter esse comportamento de estabilidade por mais tempo, viu-se a necessidade de demorar mais tempo para atingir o regime permanente.

Com os parâmetros:

Tabela 6: Parâmetros do sistema de controle extremal

Parâmetros	$r\tau = 4$	$r\tau = 5$
ω	500 rad/s	500 rad/s
Taxa de aprendizagem	0.6	0.5
Amplitude Sinal de demodulação	0.5	1
Amplitude Sinal de modulação	0.2	0.2
Frequência filtro passa-alta	0.1 rad/s	0.1 rad/s
Frequência filtro passa-baixa	1 rad/s	1 rad/s

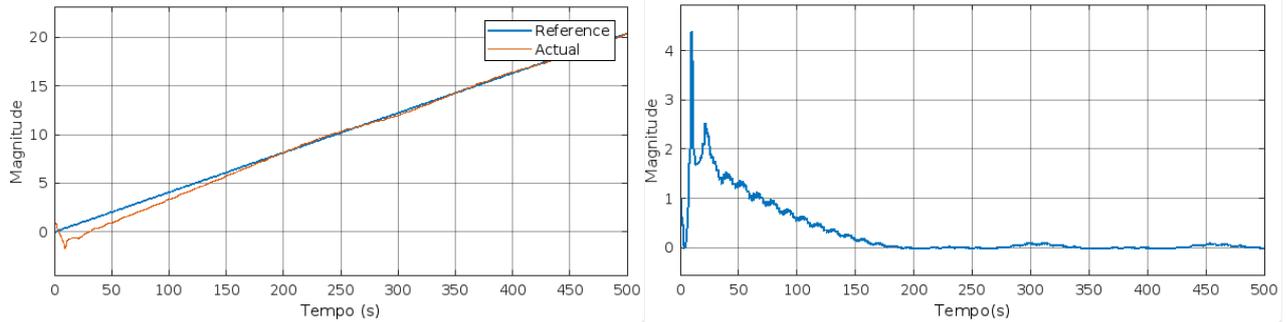
Foi obtido o resultado:



(a) Função Referência X Função Atual.

(b) Função de Custo ISE final.

Figura 43: Simulação de Controle Extremal $r\tau = 4$



(a) Função Referência X Função Atual.

(b) Função de Custo ISE final.

Figura 44: Simulação de Controle Extremal $r\tau = 5$

2 Desempenho para a Função de Custo IAE

Para essa função de custo foi realizado mesmo processo de sintonização acima.

Notou-se que uma taxa de aprendizado acima de 0.5 gerava em regime permanente um sistema que oscila bastante em torno de zero, porém com baixa amplitude, com uma taxa próxima de 0.4 o sistema iniciava estável e virava instável com alta amplitude, como apresentado abaixo. Aumentando para 0.5, o sistema voltava a aumenta a amplitude.

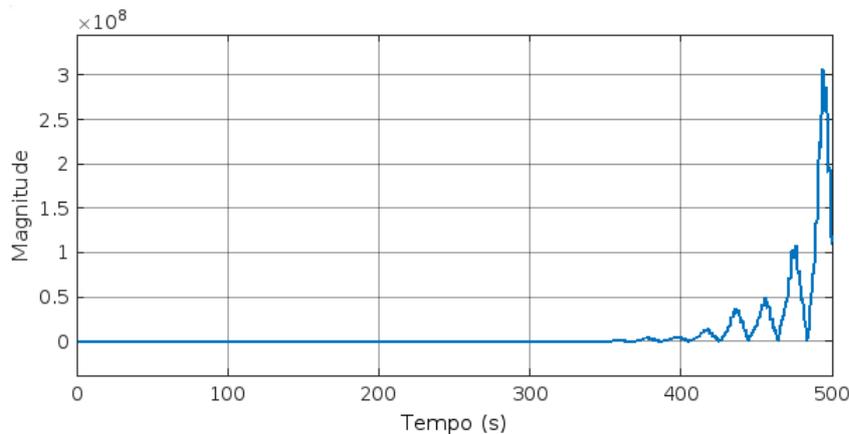


Figura 45: Função de Custo IAE com taxa de aprendizado de 0.4, $r\tau = 4$.

Aumentando e diminuindo a frequência ω o sistema se comportava como um retificador de onda crescente, logo foi mantido 500 rad/s.

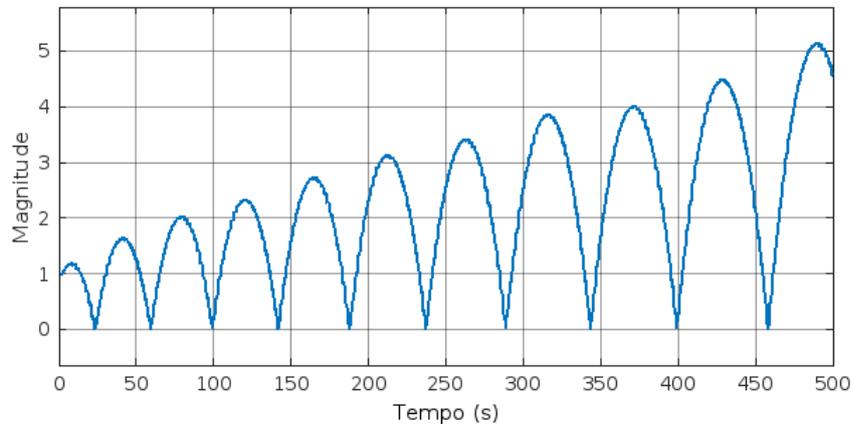


Figura 46: Função de Custo IAE com taxa de aprendizado de 0.4, $r\tau = 4$.

Aumentando a amplitude de demodulação para 2 o sistema o sistema iniciava estável e virava instável com alta amplitude, diminuindo menor que 1 ele virava retificador de onda. Mantendo 1 e mexendo na sua fase de demodulação o sistema oscilava mais ao redor de zero, por isso foi mantido sua fase em zero. Com a amplitude de 1.5 o sistema oscilava menos.

Aumentando a amplitude de modulação a oscilação em regime permanente aumentava, por isso foi mantido 0.2 sem mudança de fase.

Adicionando um filtro passa-alta de frequência 0.1 rad/s fazia com que o sistema nunca convergisse, aumentando para 1 rad/s o sistema também não convergia, como indicado abaixo. Por isso, foi decidido não adicionar tal filtro.

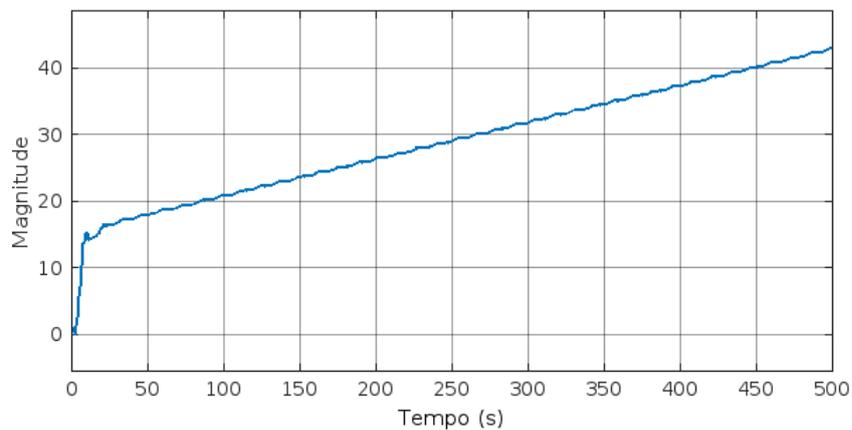


Figura 47: Função de Custo IAE com filtro passa-alta de frequência 1 rad/s, $r\tau = 4$.

Aumentando a frequência do filtro passa-baixa para 2 rad/s aumentava a frequência em regime permanente. Diminuindo para 0.1 rad/s o sistema deixava de convergir e virava oscilatório.

Com os parâmetros:

Foi obtido o resultado:

Tabela 7: Parâmetros do sistema de controle extremal

Parâmetros	$r\tau = 4$	$r\tau = 5$
ω	500 rad/s	500 rad/s
Taxa de aprendizagem	0.6	0.6
Amplitude Sinal de demodulação	1.5	0.8
Amplitude Sinal de modulação	0.2	0.1
Frequência filtro passa-alta	0 rad/s	0.1 rad/s
Frequência filtro passa-baixa	1 rad/s	2 rad/s

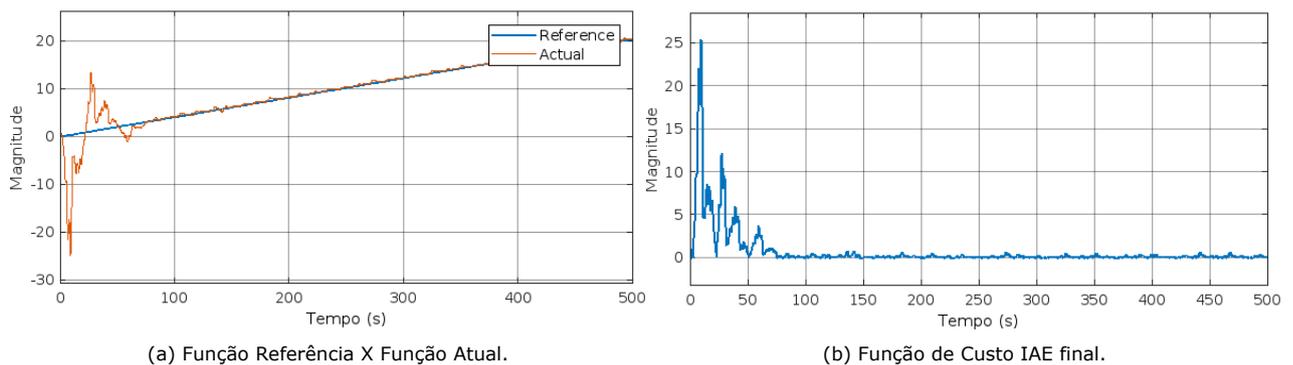


Figura 48: Simulação de Controle Extremal $r\tau = 4$

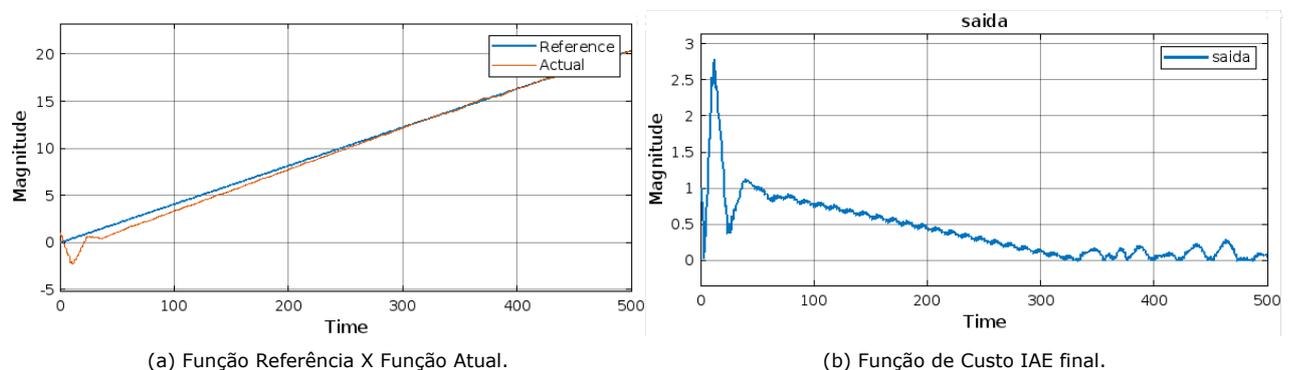


Figura 49: Simulação de Controle Extremal $r\tau = 4$

Para o caso $r\tau = 5$ o sistema apresentava alto caso de distúrbio, pequenas mudanças em seus parâmetros geravam um sistema totalmente instável. Com os parâmetros apresentados acima foi possível obter um resultado satisfatório, porém talvez aplicando outra filtragem no sinal a resposta do sistema poderia ser sistematicamente melhor.

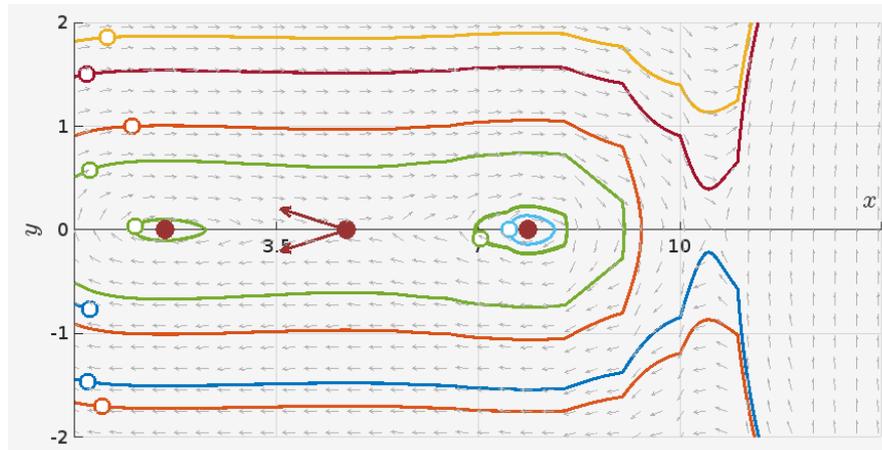
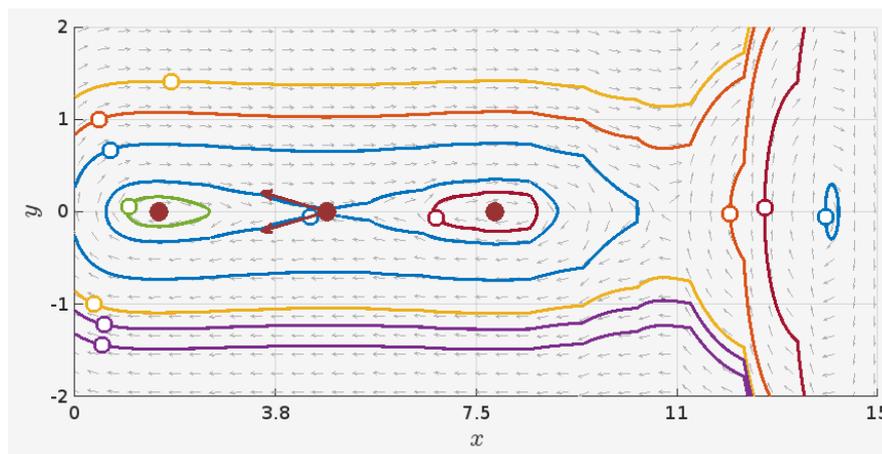
3 Phase Portrait Plotter

Com a equação de ruído e sabendo que o ganho K gerado pelo sistema de controle extremal possui um comportamento senoidal de amplitude média 30, é possível extrair as equações diferenciais ordinárias. Sendo as equações:

$$x' = e^{-r\tau} \frac{(r\tau)^x}{x!} \quad (21)$$

$$y' = \cos(x) \quad (22)$$

Foi possível definir o *Phase Portrait Plotter*. Para gerar os resultados foi utilizado o aplicativo do MATLAB™ chamado Phase Plane and Slope Field apps™.

Figura 50: *Phase Portrait Plotter* $r\tau = 4$.Figura 51: *Phase Portrait Plotter* $r\tau = 5$.

Devido a natureza não linear do sistema descrito de ruído é possível notar que as trajetórias no retrato de fase possuem características que coincidem com o comportamento esperado, como por exemplo a presença de loops fechados onde o sistema percorre padrões repetidos, ademais a presença de pontos de sela, esses pontos atuam como uma divisão, com trajetórias fluindo em direções diferentes dependendo do ponto inicial próximo à sela.

É visível notar a existência de 3 pontos de singularidade, ou seja, três pontos de estabilidade. Ao redor desses pontos temos as curvas fechadas com comportamentos cíclicos. Os pontos de sela próximo a $x = 10$, indicam que ao redor daquela região há um pólo que torna o sistema instável, o que faz com que as trajetórias tendam a infinito.

Também é visível a presença de um comportamento caótico devido a sensibilidade às condições iniciais do sistema, levando a padrões imprevisíveis e aparentemente aleatórios.

Vale ressaltar que os retratos de fase não lineares revelam o comportamento qualitativo do sistema, o que significa que mostram tendências gerais do desenvolvimento do sistema em vez de valores precisos.

7 CONCLUSÃO

O estudo de diferentes formas de analisar exoplanetas é indispensável para maximizar o desempenho e acurácia de modelos. Na literatura da área diversas pesquisas têm se empenhado na busca de soluções para otimizar a performance minimizando os custos computacionais do projeto. Neste estudo, foi avaliado um modelo matemático usando Controle Extremal e outros usando Redes Neurais e Redes Neuro-Fuzzy, e através de simulação por software, o desempenho dos sistemas modelados foram comparados, considerando métricas de avaliação comumente usados em estudos de sistemas de Aprendizado de Máquina.

Quando se analisa os desempenhos dos sistema propostos, percebe-se que ambos possuem vantagens e desvantagens perante o outro.

Na Rede Neural devido a sua natureza de "caixa preta" é difícil entender como a rede chega a uma determinada conclusão. Essa falta de transparência dificultou a depuração de erros e a interpretação dos resultados. Além disso, por mais que a Rede Neural tenha tipo resultados métricos positivos, ela ainda pode ser ajustada ainda mais para consumir menos poder computacional devido a seus hiperparâmetros. Contudo, é possível entender que a capacidade de generalização para novos dados, mesmo que não tenham sido vistos durante o treinamento, foi boa. Isso gera uma garantia de que a rede pode aprender com um conjunto de dados e aplicar esse conhecimento para resolver problemas semelhantes em cenários não vistos anteriormente.

No caso da Rede Neuro-Fuzzy, devido ao software utilizado, a base complexa de dados e a impossibilidade de melhorar os parâmetros da rede, eles acabaram perpetuando e amplificando vieses presentes nos dados de treinamento. Por isso, por mais que ela tenha dado uma acurácia impecável, seu processamento e desenvolvimento não foi um dos melhores para o tipo de projeto.

No Controle Extremal, devido sua simplicidade de implementação capaz de ajustar os parâmetros de forma iterativa para minimizar um erro predefinido gerou em um resultado bom. Visto que, isso resulta em alta precisão no controle da variável controlada, mesmo em sistemas com parâmetros incertos ou distúrbios externos. Essa técnica se destaca mais que as outras devido ao controle das variáveis e o entendimento por trás do que acontece, dado que não é usado um sistema caixa preta. Um ponto negativo, foi o tempo de simulação longo para atingir tais resultados comparado aos modelos.

Conclui-se que para a identificação de exoplanetas com uma acurácia de 0.95, é necessário o uso de Controle Extremal, visando o ruídos das imagens, mesmo que o tempo de simulação seja alto. Cabe ressaltar, que para diminuir esse tempo, basta melhorar os ajustes da modelagem do sistema. Contudo, para tal estudo o resultado foi significamente bom para análise.

Trabalhos futuros podem melhorar o sistema de Rede Neuro-Fuzzy para modelar de forma que coincide melhor para modelos astrofísicos. Além disso, pode-se desenvolver novos gráficos de análise de sistemas para entender por inteiro o comportamento dos sistemas presentes.

Referências

- [1] "No, you don't need to wake up at 6am to make something out of your life." [Online]. Available: <https://medium.com/the-coffeelicious/no-you-don-t-need-to-wake-up-at-6am-to-make-something-out-of-your-life-f2e90e633f>
- [2] "E o pequeno príncipe...tinha atitudes de cuidados com o planeta?" [Online]. Available: <https://5ano2imaculada.wordpress.com/page/2/>
- [3] "O pequeno príncipe – capítulo iv." [Online]. Available: <https://inatitude.wordpress.com/2007/11/09/o-pequeno-principe-capitulo-iv/>
- [4] "Kepler telescope declared dead after finding thousands of worlds." [Online]. Available: <https://www.cbc.ca/news/science/nasa-kepler-telescope-planet-hunting-1.4885038>
- [5] B. Guedes, "Discovering exoplanets using neural networks," 2021.
- [6] "Characteristics of the kepler space telescope." [Online]. Available: <https://keplergo.github.io/KeplerScienceWebsite/the-kepler-space-telescope.html>
- [7] "Neurônios." [Online]. Available: <https://mundoeducacao.uol.com.br/biologia/neuronios.htm>
- [8] "Model suburban commuting using subtractive clustering and anfis." [Online]. Available: <https://www.mathworks.com/help/fuzzy/model-suburban-commuting-using-subtractive-clustering.html>
- [9] "Extremum seeking control." [Online]. Available: <https://www.mathworks.com/help/slcontrol/ug/extremum-seeking-control.html>
- [10] "Diagrama de bode." [Online]. Available: <https://controlautomaticoeducacion.com/control-realimentado/1-diagrama-de-bode/>
- [11] J. D. Hartman and G. Á. Bakos, "Vartools: A program for analyzing astronomical time-series data," *Astron. Comput.*, vol. 17, pp. 1–72, 2016. [Online]. Available: <https://api.semanticscholar.org/CorpusID:63670936>
- [12] "O pequeno príncipe." [Online]. Available: <https://brasilecola.uol.com.br/literatura/o-pequeno-principe.htm>
- [13] F. Alcoforado, *A humanidade ameaçada e as estratégias para sua sobrevivência*. Editora Dialética, 2021.
- [14] N. U. Brasil, "Campanha pela # ambição climática," 2023. [Online]. Available: <https://brasil.un.org/pt-br/242958-campanha-pela-ambi%C3%A7%C3%A3o-clim%C3%A1tica#:~:text=Ainda%20%C3%A9%20poss%C3%ADvel%20limitar%20o,27%20de%20julho%20de%202023.>
- [15] "Kepler / k2." [Online]. Available: <https://science.nasa.gov/mission/kepler/>
- [16] R. V. Petrescu, R. Aversa, A. Apicella, and F. I. Petrescu, "Nasa data used to discover eighth planet circling distant star," in *Journal of Aircraft and Spacecraft Technology*, vol. 2, no. 1, 2018, pp. 19–30.
- [17] "Nasa's kepler space telescope: Final orbit." [Online]. Available: <https://exoplanets.nasa.gov/resources/2188/nasas-kepler-space-telescope-final-orbit/>
- [18] E. Martioli, "Exoplanetas: O que são e como detectá-los," *INPE*, 2006.
- [19] B. NM, "Exploring exoplanet populations with nasa's kepler mission." *Proc Natl Acad Sci U S A*, 2014.
- [20] A. K. Prithivraj G, "Identification and classification of exoplanets using machine learning techniques," *Cornell University Astrophysics, Earth and Planetary Astrophysics*, no. 2, p. 16, 2023.
- [21] M. Bugueño, F. Mena, and M. Araya, "Refining exoplanet detection using supervised learning and feature engineering," in *2018 XLIV Latin American Computer Conference (CLEI)*, no. 2, 2018, pp. 278–287.
- [22] J. Singh and N. T. Singh, "Analysis of exoplanet detection methods using machine learning and deep neural networks," in *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, 2023, pp. 1242–1247.
- [23] "What is an exoplanet?" [Online]. Available: <https://exoplanets.nasa.gov/what-is-an-exoplanet/overview/8>
- [24] "Os exoplanetas." [Online]. Available: <https://www.ufmg.br/espacodoconhecimento/exoplanetas/>

- [25] D. S. Spiegel, J. J. Fortney, and C. Sotin, "Kepler archive manual," *MAST*, 2020.
- [26] S. E. Mullally, "Structure of exoplanets," *National Academy of Sciences*, 2013.
- [27] "Lightcurvefile objects." [Online]. Available: <https://heasarc.gsfc.nasa.gov/docs/tess/LightCurveFile-Object-Tutorial.html>
- [28] "Instrumental noise in kepler and k2: Electronic noise." [Online]. Available: <https://docs.lightkurve.org/tutorials/2-creating-light-curves/2-2-kepler-noise-4-electronic-noise.html>
- [29] S. W. Hasinoff, "Photon , poisson noise," *Google Inc.* [Online]. Available: <https://people.csail.mit.edu/hasinoff/pubs/hasinoff-photon-2012-preprint.pdf>
- [30] J. C. Barbosa, "Modelagem e modelos matemáticos na educação científica," *Revista de Educação em Ciência e Tecnologia*, 2009.
- [31] P. D. . K. C. Berridge, "Model-based and model-free pavlovian reward learning: Revaluation, revision, and revelation," *Cogn Affect Behav Neurosci*, 2014.
- [32] "Ficção que parece real," 2014. [Online]. Available: https://www.correiobraziliense.com.br/app/noticia/eu-estudante/me_gerais/2014/02/17/me_gerais_interna,413180/ficcao-que-parece-real.shtml
- [33] "Uso da inteligência artificial na astronomia é tema de atividade da semana do conhecimento," 2020. [Online]. Available: <https://ufmg.br/comunicacao/noticias/atividade-da-semana-do-conhecimento-ufmg-discute-o-uso-da-inteligencia-artificial-na-astronomia>
- [34] E. E. COLZANI, "O uso da inteligência artificial no processo do trabalho: e a questão da segurança jurídica," *UNIVALI*, 2022.
- [35] A. Pinheiro, "Redes neurais artificiais," 2020. [Online]. Available: https://medium.com/@augusto_Pinheiro/redes-neurais-artificiais-133de77c7240
- [36] "Entendendo o funcionamento de uma rede neural artificial," 2018. [Online]. Available: <https://medium.com/brasil-ai/entendendo-o-funcionamento-de-uma-rede-neural-artificial-4463fcf44dd0>
- [37] USP, "Perceptron multi-camadas (mlp)." [Online]. Available: <https://sites.icmc.usp.br/andre/research/neural/mlp.htm>
- [38] I. N. da Silva, *Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas. Fundamentos Teóricos e Aspectos Práticos*. Artliber, 2016.
- [39] "Artificial neural network - perceptron." [Online]. Available: https://www.saedsayad.com/artificial_neural_network_bkp.htm
- [40] L. Askar-Zadeh, "Fuzzy sets and systems," *Symposium on System Theory XV. Polytechnic Institute of Brooklyn*, 1965.
- [41] G. C. de Oliveira Melo, "Equações diferenciais fuzzy com aplicações à epidemiologia," *CONPEEX*, 2012.
- [42] "Combinando modelos de machine learning com lógica fuzzy — parte 1s." [Online]. Available: <https://medium.com/creditas-tech/combinando-modelos-de-machine-learning-com-l%C3%B3gica-fuzzy-parte-1-b5a9f0761a5d>
- [43] "Sistemas neuro-fuzzy." [Online]. Available: https://www.maxwell.vrac.puc-rio.br/32823/32823_5.PDF
- [44] M. BURAGOHAIN, "Adaptive network based fuzzy inference system (anfis) as a tool for system identification with special emphasis on training data minimization," *Department of Electronics and Communication Engineering Indian Institute of Technology Guwahati*, 2008.
- [45] "Métricas de avaliação em machine learning: Classificação." [Online]. Available: <https://medium.com/kunumi/m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-em-machine-learning-classifica%C3%A7%C3%A3o-49340dcd198>
- [46] "As métricas mais populares para avaliar modelos de machine learning." [Online]. Available: <https://mariofilho.com/as-metricas-mais-populares-para-avaliar-modelos-de-machine-learning/>
- [47] "Quais os principais critérios da avaliação de desempenho em machine learning?" [Online]. Available: https://awari.com.br/quais-os-principais-criterios-da-avaliacao-de-desempenho-em-machine-learning/?utm_source=

blog&utm_campaign=projeto+blog&utm_medium=Quais%20os%20Principais%20Crit%C3%A9rios%20da%20Avalia%C3%A7%C3%A3o%20de%20Desempenho%20em%20Machine%20Learning#:~:text=O%20F1%20Score%20%C3%A9%20uma,positivos%20quanto%20os%20falsos%20negativos.

- [48] "What are hyperparameters? and how to tune the hyperparameters in a deep neural network?" [Online]. Available: <https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network\d0604917584a>
- [49] "Hyperparameters tuning in neural networks." [Online]. Available: <https://www.scaler.com/topics/deep-learning/neural-network-hyperparameters-tuning/>
- [50] "Activation functions in neural networks [12 types and use cases]." [Online]. Available: [https://www.v7labs.com/blog/neural-networks-activation-functions#:~:text=Tanh%20Function%20\(Hyperbolic%20Tangent\)&text=Advantages%20of%20using%20this%20activation,%2C%20neutral%2C%20or%20strongly%20positive.](https://www.v7labs.com/blog/neural-networks-activation-functions#:~:text=Tanh%20Function%20(Hyperbolic%20Tangent)&text=Advantages%20of%20using%20this%20activation,%2C%20neutral%2C%20or%20strongly%20positive.)
- [51] "Aprendizado não supervisionado em machine learning: Conceitos e aplicações." [Online]. Available: https://awari.com.br/aprendizado-nao-supervisionado-em-machine-learning-conceitos-e-aplicacoes/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Aprendizado%20N%C3%A3o%20Supervisionado%20em%20Machine%20Learning:%20Conceitos%20e%20Aplica%C3%A7%C3%B5es
- [52] "Funções de ativação." [Online]. Available: <https://matheusfacure.github.io/2017/07/12/activ-func/>
- [53] "Qual é a maneira ideal de ajustar os hiperparâmetros da rede neural?" [Online]. Available: <https://www.linkedin.com/advice/0/what-optimal-way-tune-neural-network-hyperparameters>
- [54] "O que é visão computacional?" [Online]. Available: <https://aws.amazon.com/pt/what-is-computer-vision/#:~:text=de%20vis%C3%A3o%20computacional%3F-,O%20que%20%C3%A9%20a%20vis%C3%A3o%20por%20computador%3F,las%20com%20precis%C3%A3o%20e%20efici%C3%AAncia.>
- [55] "Controle clássico." [Online]. Available: <http://sites.poli.usp.br/d/PME2472/classico.pdf>
- [56] "Automação inteligente sistemas de controle fundamentos." [Online]. Available: <https://www.cin.ufpe.br/~aluizioa/Aut-Int/Aut-Int-01-sistemas-de-controle-fundamentos.pdf>
- [57] "Instrumentação industrial: Compreendendo as malhas aberta e fechada no controle de processos." [Online]. Available: <https://www.linkedin.com/pulse/instrumenta%C3%A7%C3%A3o-industrial-compreendendo/?originalSubdomain=pt>
- [58] K. J. Åström, "Pid control," *Caltech*, pp. 1–36, 2002. [Online]. Available: <https://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom-ch6.pdf>
- [59] M. K. KARTIK B. ARIYUR, *Real-Time Optimization by Extremum-Seeking Control*. A JOHN WILEY AND SONS, INC., PUBLICATION, 2003.
- [60] "What is extremum seeking control? | learning-based control, part 1." [Online]. Available: <https://www.youtube.com/watch?v=S8aaMkqtxXw&t=4s>
- [61] A. Skafté, "An introduction to extremum-seeking control," *LTH, Sweden*, 2017.
- [62] A. a. TIAGOROUX-OLIVEIRA1, LUIZR.COSTA2, "Extremum seeking-based adaptive pid control applied to neuromuscular electrical stimulation," *Anais da Academia Brasileira de Ciências*, pp. 1–20, 2019. [Online]. Available: <https://www.scielo.br/j/aabc/a/y5Qg4pGgqzTt83NjhxhdGZ8t/?lang=en&format=pdf>
- [63] "Defining a cost function (control design and simulation module)." [Online]. Available: https://www.ni.com/docs/en-US/bundle/labview-control-design-and-simulation-module/page/lvsimconcepts/sim_c_costfunc.html
- [64] "Phase portrait." [Online]. Available: https://en.wikipedia.org/wiki/Phase_portrait
- [65] W. L. J.J E. Slotine, Jean-Jacques Slotine, *Applied Nonlinear Control*. Prentice Hall, 1990.
- [66] "Bode plots overview." [Online]. Available: <https://lpsa.swarthmore.edu/Bode/Bode.html>
- [67] NASA, "Kepler / k2," 2018.

- [68] G. Kovács, S. Zucker, and T. Mazeh, "A box-fitting algorithm in the search for periodic transits," *Astronomy and Astrophysics*, vol. 391, no. 1, pp. 369–377, Aug. 2002.
- [69] A. da Silva Teixeira, "Cálculo discreto, somatórios e progressão aritmética de ordem superior," *Universidade Federal de Uberlândia*, 2023.
- [70] E. Poço, "Integrais discretas $-r$," *Olimpíada Brasileira de Matemática*, 2007.
- [71] "K2." [Online]. Available: <https://archive.stsci.edu/missions-and-data/k2>
- [72] "Neuro-adaptive learning and anfis," 2018. [Online]. Available: <https://www.mathworks.com/help/fuzzy/neuro-adaptive-learning-and-anfis.html>
- [73] "Gas mileage prediction." [Online]. Available: <https://www.mathworks.com/help/fuzzy/gas-mileage-prediction.html>
- [74] "Adaptive noise cancellation using anfis." [Online]. Available: <https://www.mathworks.com/help/fuzzy/adaptive-noise-cancellation-using-anfis.html>
- [75] "Astropy boxleastsquares tutorial," 2018. [Online]. Available: <https://gist.github.com/hippke/9a1398995898c69038e32a937790bbb3>
- [76] "The search for new earths," 2019. [Online]. Available: <https://github.com/winterdelta/KeplerAI>
- [77] "Quick facts 3: Poisson noise," *University of Glasgow*, pp. 1–2, 2000. [Online]. Available: https://radio.astro.gla.ac.uk/old_OA_course/pw/qf3.pdf
- [78] "K2 observing." [Online]. Available: <https://keplergo.github.io/KeplerScienceWebsite/k2-observing.html>