

CENTRO UNIVERSITÁRIO DO PARÁ - CESUPA
ESCOLA DE NEGÓCIOS, TECNOLOGIA E INOVAÇÃO - ARGO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LUCAS LIMA DE ARAGÃO BARROS

**OTIMIZAÇÃO DE HIPERPARÂMETROS DE REDES NEURAIS POR INFERÊNCIA
BAYESIANA COM ENXAME DE PARTÍCULAS**

BELÉM
2023

LUCAS LIMA DE ARAGÃO BARROS

**OTIMIZAÇÃO DE HIPERPARÂMETROS DE REDES NEURAIIS POR INFERÊNCIA
BAYESIANA COM ENXAME DE PARTÍCULAS**

Trabalho de conclusão de curso apresentado à Escola de Negócios, Tecnologia e Inovação do Centro Universitário do Estado do Pará como requisito para obtenção do título de Bacharel em Ciência da Computação na modalidade ARTIGO.

Orientador: Dr. Daniel Leal Souza

Coorientador: Dr. Marco A. F. Mollinetti

BELÉM

Dados Internacionais de Catalogação-na-publicação (CIP)
Biblioteca do CESUPA, Belém – PA

Barros, Lucas Lima de Aragão.

Otimização de hiperparâmetros de redes neurais por inferência bayesiana com enxame de partículas / Lucas Lima de Aragão Barros; orientador Daniel Leal Souza, coorientador Marco A. F. Mollinetti. — 2023.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro Universitário do Estado do Pará, Belém, 2023.

- Redes neurais. 2. Hiperparâmetros. I. Souza, Daniel Leal, orient. II. Mollinetti, Marco A. F. III. Título.

23ª ed. 006.33

CDD

2023

LUCAS LIMA DE ARAGÃO BARROS

**OTIMIZAÇÃO DE HIPERPARÂMETROS DE REDES NEURAIIS POR INFERÊNCIA
BAYESIANA COM ENXAME DE PARTÍCULAS**

Trabalho de conclusão de curso apresentado à Escola de Negócios, Tecnologia e Inovação do Centro Universitário do Estado do Pará como requisito para obtenção do título de Bacharel em Ciência da Computação na modalidade ARTIGO.

Data da aprovação: 07 / 12 / 2023

Nota final aluno(a) I: 9,5

Banca examinadora



Prof(a). Daniel Leal Souza

Orientador(a) e Presidente da banca

Prof(a). Polyana Santos Fonseca Nascimento

Examinador(a) interno(a)



Prof(a). Roberto Célio Limão de Oliveira

Examinador(a) externo(a)

AGRADECIMENTOS

Agradeço ao meu pai, Augusto Barros, pelo incentivo, dedicação e suporte nos momentos mais difíceis, não me deixando desistir e insistindo. Além disso, agradeço por ter me aconselhado e proporcionado as melhores condições para alcançar meus objetivos.

Agradeço ao meu orientador Daniel Leal Souza, pelo suporte e ensinamentos essenciais neste processo. Ao meu coorientador Marco A. F. Mollinetti, pelo direcionamento e suporte direto no meu desenvolvimento e neste processo.

RESUMO

Este trabalho explora – através de técnicas de otimização hiperparamétrica – a possibilidade de melhorar a eficácia das redes neurais, que têm experimentado uma grande popularização em várias áreas. O foco é avaliar uma nova abordagem para otimização de hiperparâmetros, sendo uma das principais dificuldades no desenvolvimento dessas redes. Tradicionalmente, a definição desses hiperparâmetros ocorre de forma estocástica ou por cálculos matemáticos. O objetivo é aprimorar o processo, permitindo uma definição mais precisa dos valores, com menores perdas (*Loss*). Estas soluções baseiam-se em uma revisão da literatura sobre otimização de hiperparâmetros em Redes Neurais (e.g. CNN, *Fully Connected*), abordando conceitos fundamentais e técnicas como *Grid Search*, *Random Search* e SMBO. A metodologia inclui a escolha de ferramentas, estratégia de validação cruzada e abordagens específicas de busca, como *Gaussian Regression* ou busca em grade. Os experimentos detalhados envolvem a otimização dos hiperparâmetros das Redes Neurais, com apresentação de conjuntos de dados, configurações de modelos, protocolos de treinamento e resultados quantitativos. Apesar de suas limitações e da necessidade de mais estudos, este trabalho demonstra a viabilidade e o potencial da combinação de técnicas avançadas para otimização de hiperparâmetros em Redes Neurais, oferecendo contribuições e instigando novas direções de pesquisa na busca por maior eficácia em diversas aplicações destas redes. A discussão analisa os efeitos das configurações de hiperparâmetros, identifica limitações e aborda possíveis razões para os resultados obtidos. Por fim, a conclusão resume os principais *insights*, contribuições específicas e direções futuras, enquanto as referências garantem a fundamentação do trabalho.

Palavras-chave: Redes Neurais; Hiperparâmetros; Otimização por Enxame de Partículas; Aprendizado de Máquina, Inteligência artificial

ABSTRACT

This work explores - through hyperparameter optimization techniques - the possibility of improving the effectiveness of neural networks, which have experienced widespread popularity in various fields. The focus is to evaluate a new approach to hyperparameter optimization, which is one of the main challenges in the development of these networks. Traditionally, the definition of these hyperparameters occurs stochastically or through mathematical calculations. The goal is to refine the process, allowing a more precise definition of values, with lower losses. These solutions are based on a review of the literature on hyperparameter optimization in Neural Networks (e.g., CNN, Fully Connected), addressing fundamental concepts and techniques such as Grid Search, Random Search, and SMBO. The methodology includes the choice of tools, cross-validation strategy, and specific search approaches, such as Gaussian Regression or grid search. The detailed experiments involve optimizing the hyperparameters of Neural Networks, presenting datasets, model configurations, training protocols, and quantitative results. Despite its limitations and the need for further studies, this work demonstrates the feasibility and potential of combining advanced techniques for hyperparameter optimization in Neural Networks, offering contributions and stimulating new research directions in the search for greater efficacy in various applications of these networks. The discussion analyzes the effects of hyperparameter settings, identifies limitations, and addresses possible reasons for the obtained results. Finally, the conclusion summarizes the main insights, specific contributions, and future directions, while the references ensure the work's foundation.

Keywords: Neural Networks; Hyperparameters; Particle Swarm Optimization; Machine Learning; Artificial Intelligence.

LISTA DE SIGLAS

ANN	Redes Neurais Artificiais, do inglês <i>Artificial Neural Networks</i>
CNN	Rede Neural Convolucional, do inglês <i>Convolutional Neural Network</i>
GR	Regressão Gaussiana, do inglês <i>Gaussian Regression</i>
GS	Amostragem em Grade, do inglês <i>Grid Sampling</i>
IA	Inteligência Artificial
LHS	Amostragem por hipercubo latino, do inglês <i>Latin Hypercube Sampling</i>
ML	Aprendizado de máquina, do inglês <i>Machine Learning</i>
PSO	Otimização de Enxame de Partículas, do inglês <i>Particle Swarm Optimization</i>
SMBO	Otimização Baseada em Modelo Sequencial, do inglês <i>Sequential Model-Based Optimization</i>
SRS	Amostra Aleatória Simples, do inglês <i>Simple Random Sampling</i>
TPE	Estimador Parzen Estruturado em Árvore, do inglês <i>Tree-structured Parzen Estimator</i>

SUMÁRIO

RESUMO	6
ABSTRACT	7
1 CONTEXTUALIZAÇÃO	10
1.1 REVISÃO BIBLIOGRÁFICA	10
1.1.1 Machine Learning (ML)	10
1.1.2 Redes Neurais Artificiais (ANN)	11
1.1.3 Otimização de Hiperparâmetros	11
1.2 PROBLEMA DA PESQUISA	11
1.3 JUSTIFICATIVA	12
1.4 Objetivos	12
1.4.1 Objetivo Geral	12
1.4.2 Objetivos específicos	12
1.5 ESTRUTURA DO TRABALHO	13
2 ARTIGO	13
2.1 INTRODUÇÃO	13
2.2 METODOLOGIA DA PESQUISA	14
2.2.1 Processo de Treinamento e Avaliação de Modelos de Aprendizado de Máquina	15
2.2.2 Latin Hypercube Sampling (LHS)	22
2.2.3 Simple Random Sampling (SRS)	23
2.2.4 Grid Sampling (GS)	24
2.2.5 Particle Swarm Optimization (PSO)	25
2.2.5 Otimização Baseada em Modelo Sequencial (SMBO)	27
2.3 RESULTADOS	29
2.3.1 Fully Connected ANN	29
2.3.2 Convolutional Neural Network (CNN)	31
2.4 DISCUSSÃO	34
2.5 CONCLUSÃO/CONSIDERAÇÕES FINAIS	35
3 REFERÊNCIAS BIBLIOGRÁFICAS	36

1 CONTEXTUALIZAÇÃO

1.1 Revisão Bibliográfica

1.1.1 Machine Learning (ML)

A Inteligência Artificial (IA) é um campo da ciência da computação que se concentra no desenvolvimento de sistemas computacionais capazes de realizar tarefas que, se realizadas por seres humanos, normalmente, exigiria inteligência humana. Isso inclui a capacidade de aprender, raciocinar, resolver problemas, tomar decisões e compreender linguagem natural. A IA busca criar algoritmos e modelos que permitam às máquinas simular comportamentos inteligentes e autônomos (SHAMBU, 2023).

A ML é uma vertente essencial da IA e concentra-se no desenvolvimento de algoritmos e modelos que capacitam sistemas a aprender padrões e tomar decisões com base em dados. Ao invés de depender de programação explícita, os modelos de ML são alimentados com dados, permitindo que aprendam e realizem previsões ou tomem decisões automaticamente.

Dentro das categorias principais de aprendizado de máquina, o aprendizado supervisionado envolve treinar modelos com dados rotulados, capacitando-os a fazer previsões em novos dados. No aprendizado não supervisionado, modelos exploram dados não rotulados para descobrir padrões intrínsecos, enquanto o aprendizado por reforço implica que um agente tome decisões em um ambiente para maximizar recompensas, aprendendo por tentativa e erro.

Sistemas baseados em ML desempenham um papel fundamental em aplicações de IA, tais como reconhecimento de padrões, processamento de linguagem natural, visão computacional, dentre outros. Em uma sociedade cada vez mais dependente de soluções baseadas em Inteligência Artificial, modelos de ML se posicionam como ferramentas cruciais em diversas áreas, melhorando seu desempenho à medida que são expostos a mais dados, consequentemente impulsionando avanços nos mais diversos setores, como diagnóstico médico, recomendações de conteúdo, etc..

1.1.2 Redes Neurais Artificiais (ANN)

As ANNs são modelos computacionais que se inspiram, em particular, na estrutura e no funcionamento do sistema nervoso do cérebro humano, para tarefas de processamento de informações. Elas fazem parte do campo da Inteligência Artificial e se destacam por sua capacidade de aprender e generalizar a partir de dados (TOWARDS DATA SCIENCE, 2019).

As ANNs são compostas por unidades de processamento denominadas neurônios artificiais ou perceptrons, organizados em camadas interconectadas. Essas conexões, denominadas sinapses, possuem pesos que são ajustados durante o treinamento da rede. O processo de aprendizado envolve a otimização desses pesos para que a rede possa realizar tarefas específicas, como reconhecimento de padrões, classificação de dados, previsão e muito mais.

Uma característica marcante das ANNs é a capacidade de aprendizado. Permite que elas se adaptem a diferentes tipos de dados e problemas. São especialmente eficazes em tarefas onde padrões complexos e não-lineares precisam ser identificados. Além disso, abrangem diversas áreas, incluindo visão computacional, processamento de linguagem natural, robótica, diagnóstico médico, previsão de séries temporais, entre outras. Sua flexibilidade e desempenho são valiosos em muitos campos da ciência e da indústria.

1.1.3 Otimização de Hiperparâmetros

Hiperparâmetros em ANN são configurações predefinidas que controlam aspectos da arquitetura, do treinamento e da otimização da rede. Eles não são aprendidos diretamente, mas desempenham um papel crucial no desempenho e comportamento da ANN. Exemplos incluem taxa de aprendizado, número de épocas, arquitetura da rede e outros. A escolha cuidadosa e ajuste adequado dos hiperparâmetros é essencial para o sucesso da ANN em tarefas de ML (CLAESEN; DE MOOR, 2023).

Muitos algoritmos de visão computacional dependem de configurações que são tipicamente ajustadas manualmente durante a avaliação do algoritmo para um conjunto específico de dados (BERGSTRA; YAMINS, 2013).

1.2 Problema da Pesquisa

A hiperparametrização de redes neurais é uma tarefa crítica no desenvolvimento de modelos de ANN, que apresenta desafios. Problemas comuns incluem sobreajuste

(overfitting) e sub ajuste (underfitting) devido a escolhas inadequadas de hiperparâmetros, busca ineficaz de hiperparâmetros, espaços de hiperparâmetros mal definidos, interdependência de hiperparâmetros e dificuldade de interpretação. Para enfrentar esses desafios, são necessárias abordagens como busca sistemática, validação cruzada e otimização bayesiana. A seleção adequada de hiperparâmetros é fundamental para obter modelos de redes neurais eficazes e generalizáveis. Explicando melhor, o problema aqui abordado considera eficácia como: quanto menor a perda (Loss) maior a eficácia. Nesse sentido, a pergunta que se pretende ver respondida neste estudo é: o efeito da combinação das técnicas atuais na otimização de hiperparâmetros é mais eficaz, em comparação com o uso das mesmas técnicas de forma isolada?

1.3 Justificativa

É possível que se possa utilizar outras abordagens, mas neste trabalho escolheu-se combinar as técnicas descritas no parágrafo seguinte, na hipótese de que o resultado seja a maior eficácia em relação ao uso de outras técnicas de forma isolada.

Foram escolhidas as seguintes técnicas: (1) Particle Swarm Optimization (PSO), empregado na otimização de hiperparâmetros devido à sua capacidade de explorar globalmente o espaço de busca de maneira simplificada; e (2) o Sequential Model-Based Optimization (SMBO) (BO et al, 2022) também chamada de Otimização Bayesiana (BERGSTRA et al, 2011), em português Otimização Baseada em Modelo Sequencial, em razão da sua contribuição probabilística na otimização.

Por essas características, resolveu-se utilizar o PSO e o SMBO, para testar se, combinados, aprimoram a eficácia em modelos de ML.

1.4 Objetivos

1.4.1 Objetivo Geral

Investigar a eficácia de novos métodos na busca de hiperparâmetros em redes neurais por meio de avaliações práticas.

1.4.2 Objetivos específicos

- Identificar e revisar métodos convencionais e novos métodos de otimização de hiperparâmetros.
- Realizar simulações utilizando diferentes combinações dos métodos de otimização em diferentes arquiteturas de redes neurais artificiais.

- Analisar e comparar os resultados das simulações para avaliar a eficácia e adequação dos métodos.

1.5 Estrutura do Trabalho

O primeiro capítulo consiste na introdução da temática de otimização em ANN, e é apresentado um referencial teórico, o problema de pesquisa e a justificativa que nortearam o estudo e seus objetivos. No segundo capítulo apresenta-se o artigo, com: introdução; metodologia, esclarecendo a escolha das técnicas exploradas com suas respectivas explicações; resultados, obtidos a partir dos testes de ML; discussão desses mesmos resultados; e conclusão com as considerações finais.

2 ARTIGO

2.1 Introdução

A rápida adoção da IA pelo mundo tem sido uma tendência extraordinária nas últimas décadas, impulsionada pelo avanço tecnológico e pela busca por eficácia e inovação em diversos setores. A IA está transformando radicalmente a maneira como empresas, governos e indivíduos abordam problemas complexos, desde diagnósticos médicos mais precisos, otimização de cadeias de suprimentos, assistência virtual ao cliente, até aplicações na exploração do petróleo (WANG et al, 2023). Essa aplicação crescente da IA teve influência decisiva na escolha do tema deste trabalho que busca contribuir com esse crescimento a partir do estudo da otimização de técnicas de ML, que possui potencial de aceleração da velocidade, redução de custos e aumento da eficácia no mencionado aprendizado.

A otimização de hiperparâmetros em ANNs emerge como um tema crucial na Ciência da Computação, particularmente pela crescente necessidade de refinar a eficácia dos modelos de ML. Com a disseminação das ANNs em diversas aplicações, desde reconhecimento de padrões até previsões complexas, uma configuração otimizada dessas redes torna-se imperativa.

Vale ressaltar novamente que o estudo se restringe à verificação da eficácia, mas há também um amplo terreno a ser investigado no terreno da eficácia na busca de reduzir os custos computacionais e temporais associados ao treinamento e inferência.

No contexto específico deste trabalho, a pesquisa aventurou-se no campo da hiperparametrização via PSO (Otimização de Enxame de Partículas) (JADERBERG *et al*, 2017) e SMBO (BO *et al*, 2022), em português Otimização Baseada em Modelo Sequencial, também chamada de Otimização Bayesiana (BERGSTRA *et al*, 2011), um método probabilístico. Dentre as diversas técnicas para a configuração de hiperparâmetros, este estudo utiliza a *Gaussian Regression* (GR), associada à função de Covariância Matérn (HANDCOCK *et al*, 2023).

Estes mecanismos, em conjunto, visam não apenas entender a relação entre os hiperparâmetros e como se comportam em relação à métrica de desempenho adotada, a perda (*Loss*).

Apesar do vasto corpo de literatura existente, ainda há lacunas na eficácia da otimização de hiperparâmetros e já existem estudos de algoritmos de busca baseados em população (PSO) e Otimização Bayesiana (SMBO).

Além disso, muitos métodos tradicionais de otimização exigem avaliações repetidas e custosas, tornando o processo ineficiente em termos de tempo e recursos, como apontam artigos científicos relacionados ao tema encontrados em conferências de aprendizado de máquina e otimização, como o "NeurIPS" (Conference on Neural Information Processing Systems) e "ICML" (International Conference on Machine Learning), onde pesquisadores compartilham suas descobertas, mas este não é o foco deste trabalho.

Neste estudo, a abordagem proposta alavanca técnicas estatísticas e otimização bayesiana para prever e ajustar hiperparâmetros na busca de reduzir as perdas. Utilizando amostras geradas através do Latin Hypercube Sampling (LHS) e do Simple Random Sampling (SRS), o trabalho se propõe a avaliar a eficácia da combinação da GR e do PSO na otimização dos hiperparâmetros de modelos de ANNs, como uma rede neural totalmente conectada e Rede Neural Convolutiva, do inglês *Convolutional Neural Network* (CNN). A hipótese central é que esta combinação pode fornecer resultados superiores em comparação aos métodos tradicionais, reduzindo as perdas.

2.2 Metodologia da Pesquisa

Este artigo é baseado em uma pesquisa exploratória que levou em consideração as informações obtidas nos acervos on-line e repositórios acadêmicos, com materiais bibliográficos acerca das temáticas de otimização de hiperparâmetros, para realizar

experimentos, atingir seus objetivos geral e específicos e responder a pergunta inicial: há melhora da eficácia do processo de hiperparametrização com a combinação de técnicas de PSO e SMBO?

Foram estabelecidos critérios de busca das respostas a partir dessa pergunta para limitar o universo da pesquisa bibliográfica tanto em horizonte temporal quanto em palavras-chaves, com o objetivo de evitar dispersão conforme se descreve no Quadro 1 abaixo.

Quadro 1 - Dados da Pesquisa

Tema	Parâmetros
Pergunta inicial da pesquisa	A otimização com PSO e SMBO é viável?
Termos de busca	<i>Particle Swarm Optimization, Gaussian Regression, Bayesian Optimization, Latin Hypercube Sampling, overview, Multi Layer Perceptron, Tree-structured Parzen Estimator, Sequential Model-Based Optimization, Artificial Neural Networks</i>
Ano de publicação	De 2019 a novembro/2023
Idioma	Português e Inglês

Fonte: Elaborada pelo autor (2023)

2.2.1 Processo de Treinamento e Avaliação de Modelos de Aprendizado de Máquina

Descreve-se abaixo o passo a passo do processo operado nesta pesquisa com um fluxo contínuo de etapas que proporciona uma visão ordenada do processo de otimização de hiperparâmetros de ANNs, visando explorar a viabilidade do método, culminando na avaliação estatística dos resultados para análise e comparação.

2.2.1.1 Configuração Inicial e definição de Modelos.

Os experimentos foram feitos na linguagem de programação Python, utilizando as bibliotecas que são ferramentas que facilitaram o desenvolvimento do trabalho, nomeadamente: Scipy, TensorFlow, Sklearn, Pandas, Numpy, Hyperopt.

No Quadro 2 a seguir, resume-se cada uma destas bibliotecas:

Quadro 2 - Resumo de Bibliotecas de Ciência de Dados e Aprendizado de Máquina

Nome	Descrição Resumida	Referência Bibliográfica
Scipy	Biblioteca de Python para computação científica, oferecendo módulos para otimização, álgebra linear, integração, interpolação, funções especiais, FFT (Transformada Rápida de <i>Fourier</i>), processamento de sinais e imagens, e outras tarefas comuns em ciência e engenharia.	VIRTANEN, P., et al. (2020).
TensorFlow	Biblioteca de código aberto para aprendizado de máquina e inteligência artificial, focada em treinamento e inferência de redes neurais profundas.	ABADI, M., et al. (2016).
Sklearn	Também conhecida como scikit-learn, é uma biblioteca para aprendizado de máquina em Python, oferecendo uma variedade de ferramentas para modelagem estatística, incluindo classificação, regressão, clustering e redução de dimensionalidade.	PEDREGOSA, F., et al. (2011).
Pandas	Biblioteca que oferece estruturas de dados e ferramentas de análise de dados de alto desempenho e fáceis de usar para Python.	MCKINNEY, W. (2010).
Numpy	Biblioteca fundamental para computação científica com Python, fornecendo suporte para <i>arrays</i> (estruturas de dados) multidimensionais e uma ampla coleção de funções matemáticas para operações com esses arrays.	HARRIS, C. R., et al. (2020).
Hyperopt	Biblioteca para otimização de hiperparâmetros, utilizando algoritmos como busca bayesiana, árvores de parzen e outros métodos de otimização.	BERGSTRA, J., et al. (2013).

Fonte: Elaborado pelo autor (2023)

Escolheu-se, para esta pesquisa, as seguintes redes neurais: *Convolutated Neural Network* (CNN), *Fully Connected - ANN* (*Artificial Neural Network*).

Resume-se a seguir as características das redes neurais adotadas no experimento.

CNN: Redes neurais especializadas em processar dados com uma grade-topologia, como imagens. Utilizam camadas convolucionais que aplicam filtros para detectar características locais em dados de entrada. Amplamente usadas em visão computacional e processamento de imagem (LE CUN, Y., BENGIO, Y., & HINTON, G., 2015).

Os hiperparâmetros estudados dentro da CNN foram: número de neurônios na primeira e na segunda camada de convolução, o tamanho dos filtros usados em cada uma das duas camadas de convolução, número de neurônios no primeiro e no segundo *Hidden Layer* (camada oculta), otimizador utilizado no treinamento da ANN, taxa de aprendizado e tamanho do lote (*Batch Size*) utilizado em uma iteração

ANN: Redes neurais onde cada neurônio em uma camada está conectado a todos os neurônios na próxima camada, formando uma arquitetura densamente conectada. Adequadas para tarefas gerais de aprendizado de máquina, incluindo classificação e regressão (GOODFELLOW, I., BENGIO, Y., & COURVILLE, A., 2016).

Não foi possível realizar testes em técnicas como Adafactor (SHAZEER *et al*, 2018), pois faz parte do seu cálculo deletar as 2 últimas dimensões, reduzindo os valores na função ao ponto de não conseguir influenciar no resultado de maneira expressiva.

Os hiperparâmetros estudados dentro da ANN foram: número de neurônios no primeiro e no segundo *Hidden Layer* (camada oculta), otimizador utilizado no treinamento da ANN, taxa de aprendizado e tamanho do lote (*Batch Size*) utilizado em uma iteração.

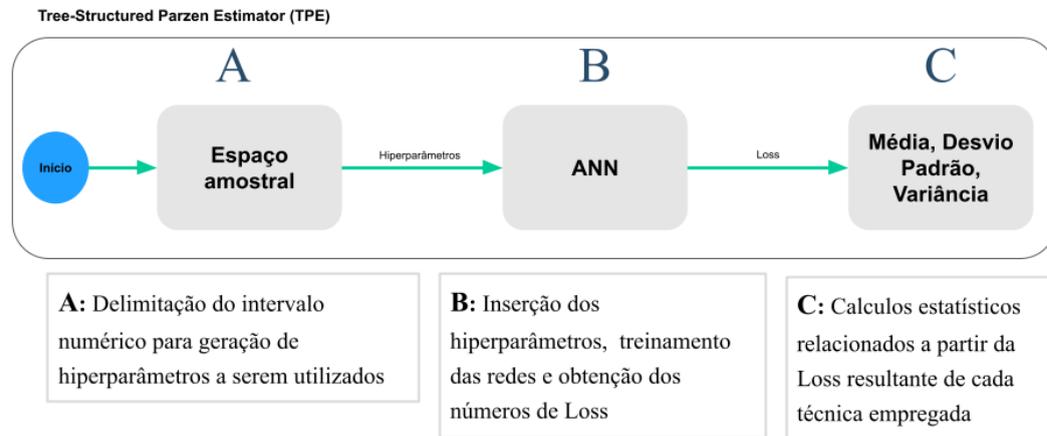
Diferentes conjuntos de hiperparâmetros, foram gerados, usando as técnicas de amostragem *Latin Hypercube Sampling* (LHS), item 2.2.2, *Simple Random Sampling* (SRS) item 2.2.3 e *Grid Sampling* (GS) item 2.2.4, delimitando o espaço amostral e definindo as configurações das ANN.

2.2.1.2 Treinamento das Redes Neurais e Obtenção de *Loss*

Realizou-se 13 conjuntos de 100 treinamentos na linha de treinamento com *Tree-Structured Parzen Estimator* (TPE), descrita no item 2.2.5.2 mais abaixo. As técnicas mencionadas no parágrafo anterior (LHS, SRS e GS) não foram necessárias pois o algoritmo do TPE gera os próprios hiperparâmetros, realiza todo o processo de amostragem dentro do mesmo espaço amostral, treinamento e otimização e, para efeito de equilíbrio das proporções, foram efetivados somente 13 conjuntos de 100 treinamentos.

A Figura 1 abaixo mostra o fluxo TPE aplicado após serem completados os processos de otimização.

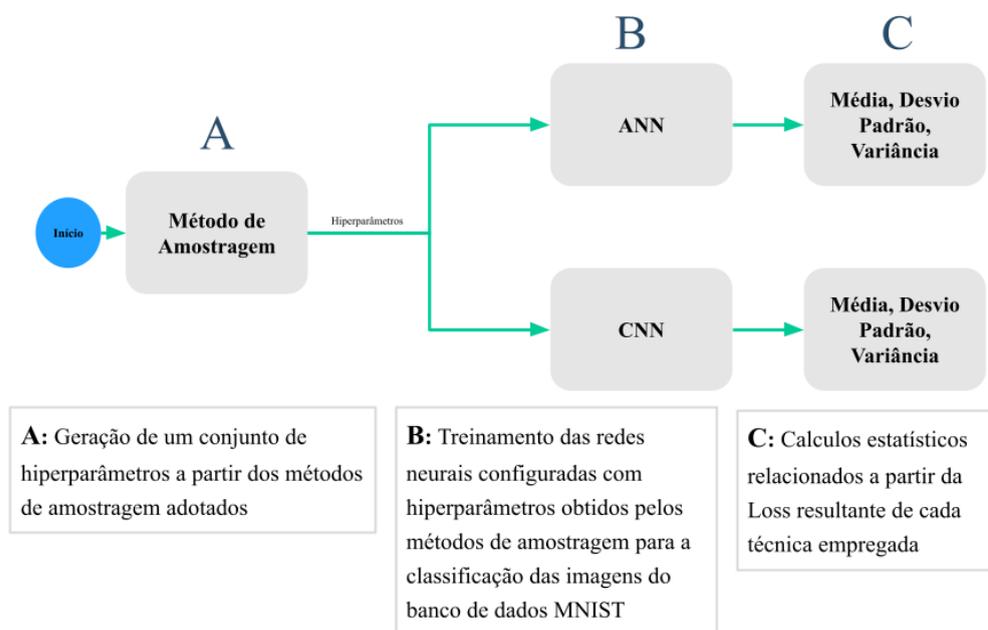
Figura 1 - Fluxo TPE



Fonte: Elaborada pelo autor (2023)

O conjunto de dados escolhido para treinamento das ANNs foi o MNIST, por tratar-se de um grande banco de dados de dígitos manuscritos comumente utilizado (HIRATA *et al*, 2023) para treinamento de sistemas de processamento de imagens (BYERLY, *et al*, 2021), objeto de de estudos reconhecidos no meio acadêmico, bem como, pela sua simplicidade.

Os processos de treinamentos utilizando as amostras geradas pelos métodos SRS, LHS, GS, também foram realizados com o banco de dados MNIST, com as definições descritas no item 2.2.1.1 ilustrados pela Figura 2, abaixo:

Figura 2 - Fluxo Hiperparâmetros e *Loss*

Fonte: Elaborada pelo autor (2023)

Com os treinamentos realizados, obteve-se a perda (*Loss*) que representa a diferença entre o valor de saída da rede neural e o esperado para determinada entrada. A *Loss* foi o parâmetro utilizado para avaliar e comparar a eficácia de uma ANN. Lembrando que este estudo não busca o melhor valor existente, mas sim o melhor valor num determinado universo. Salientando, que não foram medidos tempos de execução.

Foi extraído o valor da menor *Loss* dentre os 1000 treinamentos de cada um dos 20 conjuntos. De posse dessas 20 menores *Loss*, foram realizados os cálculos de média, variância e desvio padrão, para análise e comparação.

2.2.1.3 Interpolação com Gaussian Regression inicial

Com os conjuntos de hiperparâmetros e suas respectivas *Loss*, foi realizada a interpolação dos respectivos dados com a técnica Gaussian Regression, componente da otimização Bayesiana, a fim de delimitar uma área em torno do menor resultado de *Loss* obtido, estimando a covariância.

Após a delimitação da área, a Gaussian Regression realizou uma predição linear de novos conjuntos de hiperparâmetros e calculou-se a melhora esperada em relação ao conjunto

anterior, empregando-se a função de aquisição Expected Improvement (EI), no sentido de maximização.

2.2.1.4 Execução conjunta de PSO e otimização Bayesiana

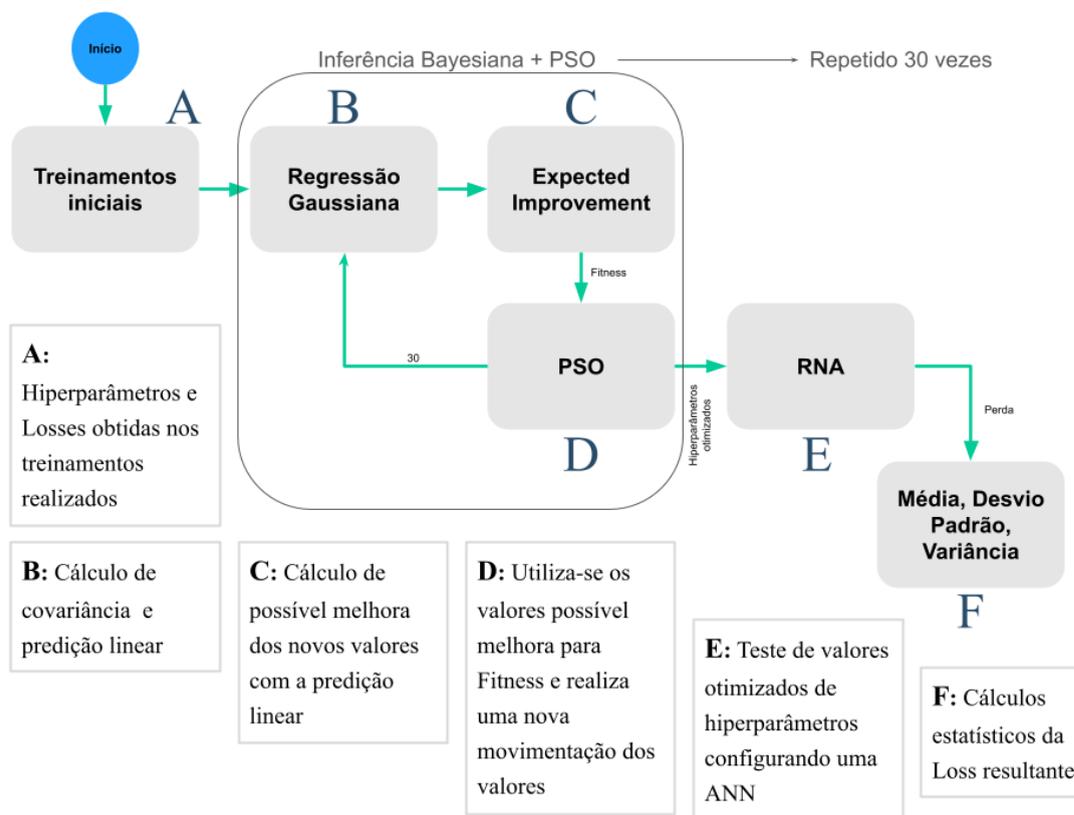
Os dados obtidos através da predição linear foram utilizados como novos conjuntos de hiperparâmetros e seus respectivos valores de EI, como dados de entrada para o PSO.

Foram feitos múltiplos testes, com diferentes quantidades de novos hiperparâmetros, assim utilizando diferentes quantidades de partículas no PSO, pois as tais recebiam os hiperparâmetros como característica e definia a sua respectiva fitness com o correspondente valor de EI. Com base nessas definições, o PSO realizou 30 iterações. Em cada iteração empregou-se o processo de otimização Bayesiana, para obtenção de uma nova fitness correspondente a cada conjunto de amostra.

Este processo se repetiu por 20 iterações, em cada conjunto de hiperparâmetros utilizado para treinamento das respectivas redes neurais, por conseguinte obtendo valores otimizados de Loss.

A Figura 3 abaixo expressa o fluxo concatenado dos processos descritos nos itens 2.2.1.3 e 2.2.1.4:

Figura 3 - PSO e Otimização Bayesiana



Fonte: Elaborada pelo autor (2023)

O fluxo da Figura 2, acima, representa os processos realizados com os métodos de amostragem Gaussian Regression, PSO e otimização Bayesiana.

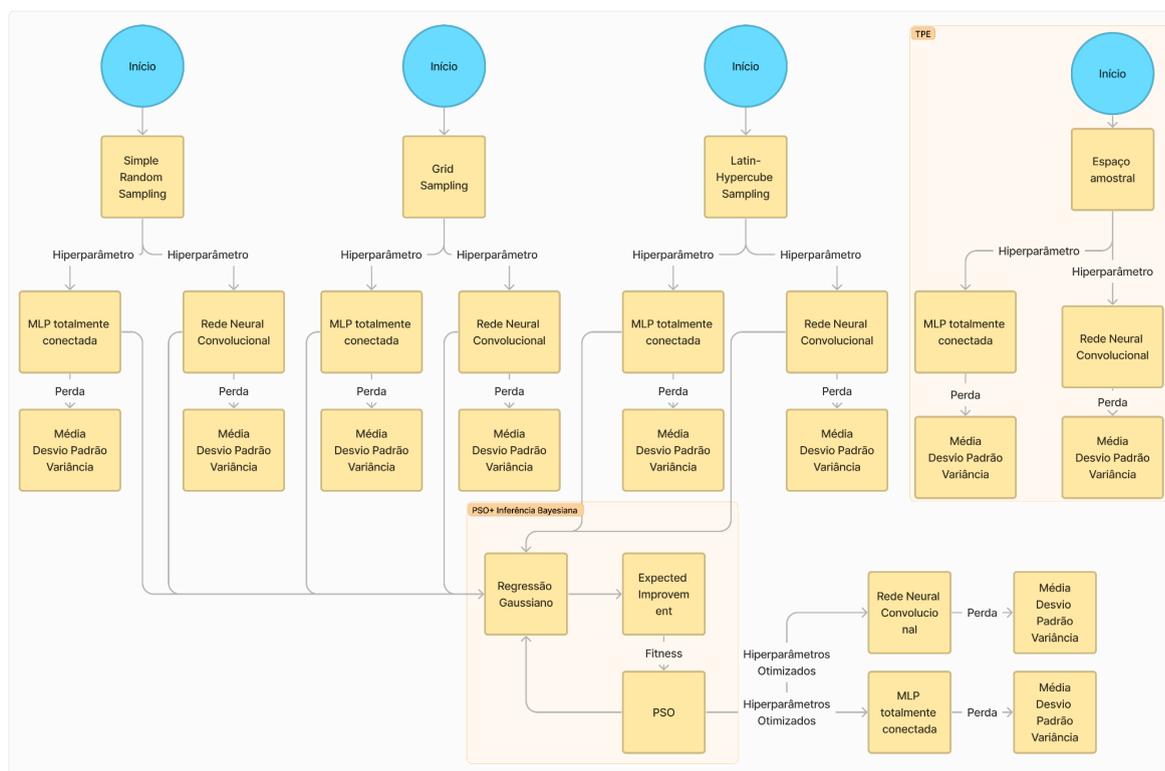
2.2.1.5 Avaliação Estatística dos Resultados

No final, utilizou-se o cálculo da média, variância e desvio padrão dos valores otimizados de *Loss*, para comparação dos resultados obtidos com os anteriores, permitindo uma análise estatística completa do desempenho das técnicas de otimização.

Este processo sistemático permitiu comparar como diferentes técnicas de amostragem de hiperparâmetros afetam o desempenho dos modelos de ML, através da análise da Média, Variância e do Desvio Padrão.

Finalmente, os fluxos individuais mencionados anteriormente estão agrupados na Figura 4 abaixo, que representa o processo completo executado durante a pesquisa:

Figura 4 - Fluxo completo



Fonte: Elaborada pelo autor (2023)

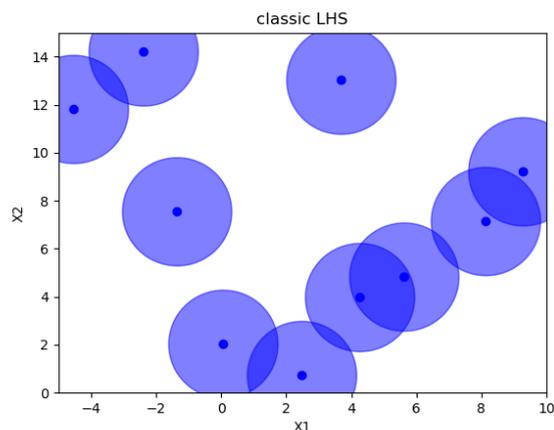
A seguir, descreve-se as técnicas utilizadas para operacionalizar esta pesquisa.

2.2.2 Latin Hypercube Sampling (LHS)

A *Latin Hypercube Sampling* (LHS) é uma técnica estatística que aqui foi utilizada por permitir a simulação computacional e experimentos para amostrar em intervalos iguais em um espaço multidimensional.

Como exemplifica a Figura 5, abaixo.

Figura 5 - Exemplo de LHS



Fonte:

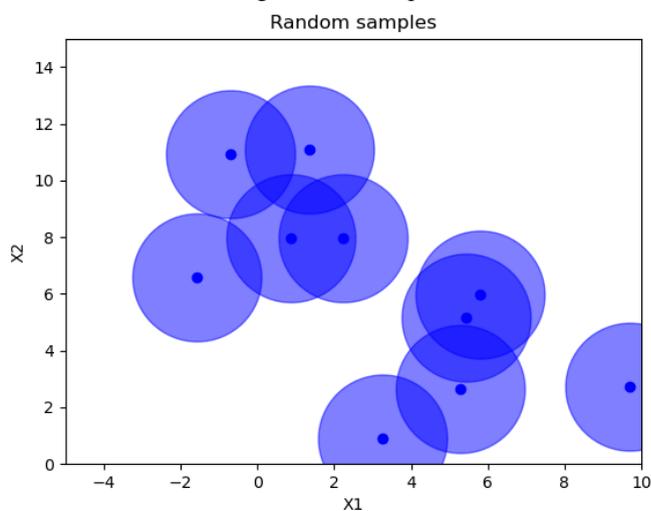
https://scikit-optimize.github.io/stable/auto_examples/sampler/initial-sampling-method.html#random-sampling
Acessada em: 22, nov 2023.

A LHS divide o espaço de parâmetros em estratos e seleciona uma amostra única em cada estrato, assegurando que a amostra seja representativa e não redundante. Isso ajuda a economizar recursos computacionais e a obter resultados mais precisos em análises de sistemas complexos com numerosas variáveis de entrada (SCIENCE DIRECT, 2023).

2.2.3 Simple Random Sampling (SRS)

A Simple Random Sampling (SRS) (SINGH, 2003), aqui empregada, é o método mais simples e comum de selecionar uma amostra. Nele a amostra é selecionada unidade por unidade, com igual probabilidade de seleção para cada unidade em cada sorteio. Ilustrado na Figura 6, abaixo.

Figura 6 - Exemplo de SRS



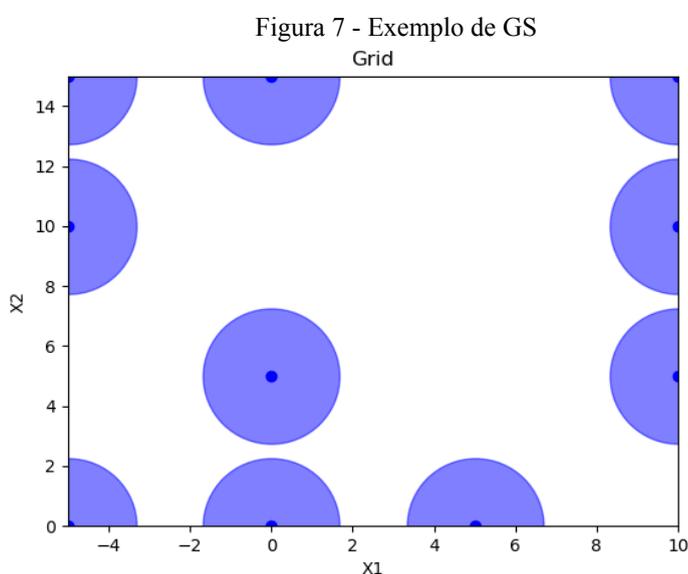
Fonte:

https://scikit-optimize.github.io/stable/auto_examples/sampler/initial-sampling-method.html#random-sampling
Acessada em: 22, nov 2023.

Em outras palavras, a amostragem aleatória simples é um método de selecionar uma amostra s de n unidades de uma população Ω de tamanho N dando igual probabilidade de seleção a todas as unidades. É um esquema de amostragem no qual todas as combinações possíveis de n unidades podem ser formadas a partir da população de N unidades com a mesma chance de seleção (BERGSTRA; BENGIO, 2012).

2.2.4 Grid Sampling (GS)

O Grid Sampling (GS), ou amostragem em grade, é uma abordagem sistemática de coleta de dados que divide a área de interesse em uma grade regular e coleta amostras em pontos específicos dessa grade. Vide exemplificação na Figura 7, abaixo.



Fonte:

https://scikit-optimize.github.io/stable/auto_examples/sampler/initial-sampling-method.html#random-sampling
Acessada em: 22, nov 2023.

Essa técnica é comumente utilizada em pesquisas espaciais, experimentos científicos e análises de dados geoespaciais. Ao distribuir pontos de amostragem de maneira uniforme em uma grade, o *Grid Sampling* visa garantir uma representação abrangente e equitativa de uma determinada área, minimizando viés e permitindo uma análise mais precisa e generalizada dos dados coletados. Essa estratégia é particularmente útil em situações em que a heterogeneidade espacial é um fator importante a ser considerado, contribuindo para a obtenção de resultados mais representativos e confiáveis (WU *et al*, 2021).

2.2.5 Particle Swarm Optimization (PSO)

A PSO (KENNEDY; EBERHART, 1995) foi utilizada nesta pesquisa como técnica na qual um conjunto de entidades simples chamadas partículas, é posicionado no espaço de busca de um problema ou função específica. Cada partícula avalia a função objetivo em sua posição atual. Em seguida, cada partícula determina seu próximo movimento através do espaço de busca combinando informações do seu histórico pessoal, incluindo sua melhor posição anterior, com informações de um ou mais membros do grupo, além de introduzir algumas perturbações aleatórias. A próxima iteração ocorre após todas as partículas terem sido movidas. Com o tempo, o grupo todo, como um bando de pássaros na busca coletiva por alimento, tende a se aproximar do ótimo da função de aptidão (POLI *et al*, 2007).

PSO representa potenciais soluções como partículas em um espaço multidimensional, atribuindo a cada partícula uma posição e velocidade, estão sendo calculadas pelas equações abaixo nas Equações 1 (velocidade) e 2 (posição).

$$V_{[i][j]}(t + 1) = wV_{[i][j]}(t) + R_1C_1(P_{[i][j]}(t) - X_{[i][j]}(t)) + R_2C_2(P_{[g][j]}(t) - X_{[i][j]}(t)) \quad (1)$$

$$X_{[i][j]}(t + 1) = X_{[i][j]}(t) + V_{[i][j]}(t + 1) \quad (2)$$

As variáveis na equação são: Iteração atual da partícula (t); Fator de inércia (w); Índice da partícula (i); índice da variável (j); Valor atual de velocidade ($V_{[i][j]}(t)$); Novo valor de velocidade ($V_{[i][j]}(t + 1)$); Valor atual de posição ($X_{[i][j]}(t)$); Novo valor de posição ($X_{[i][j]}(t + 1)$); Melhor local ($P_{[i][j]}(t)$); Melhor global ($P_{[g][j]}(t)$); Coeficientes de aceleração individual e social (C_1 e C_2); Números aleatórios uniformes entre de 0 até 1 (R_1 e R_2).

Incorporado posteriormente no algoritmo PSO original por (SHI, Yuhui; EBERHART, Russell. 1998) (SHI, Yuhui *et al.*, 2001), o fator de inércia ($w(t)$) tem como principal objetivo a melhoria na taxa de convergência do resultado, ou seja, a capacidade com que as partículas do enxame encontram a solução ótima, ou algum resultado muito próximo dessa solução. (SOUZA, Daniel., 2021). Com a Equação 3, espera-se obter valores entre 0,5 e 1.

$$w(t + 1) = 0,5 + \frac{rand}{2} \quad (3)$$

As partículas iterativamente ajustam suas posições com base na avaliação de uma função objetivo, que é calculada pela técnica Gaussian Regression e EI, que estima a qualidade da solução. A interação entre as partículas é facilitada pela comunicação direta e indireta, permitindo que o enxame explore o espaço de busca em direção a regiões de soluções ótimas. O PSO tem sido amplamente aplicado em problemas de otimização complexos em diversas áreas (B. A. S. Emambocus. et al., 2023), incluindo engenharia, ciência da computação e aprendizado de máquina (Li. Y., et al., 2021).

Além disso, foi configurada uma técnica de *Damping* ou “amortecimento”, que se refere ao processo de reduzir a intensidade, amplitude ou oscilação de um sistema físico. Em diversos contextos, o amortecimento é utilizado para mitigar ou controlar movimentos, vibrações ou flutuações excessivas.

Ver figura 8 abaixo:

Figura 8 - Algoritmo 1: PSO Global clássico com fator de inércia

```

1 for i ← 0 to (QUANTIDADE DE PARTÍCULAS-1) do
2     for j ← 0 to (QUANTIDADE DE VARIÁVEIS-1) do
3         Inicializar os valores de velocidade;
4         Inicializar os valores de posição;
5     Avaliar partícula (Fitness);
6     Inicializar o melhor local da partícula ( $P_{[i][j]}(t)$ );
7     Obter o melhor global do enxame ( $P_{gj}(t)$ );
8 for iter ← 1 to (QUANTIDADE DE ITERAÇÕES-1) do
9     Atualizar fator de inércia (Equação 2.12 ou 2.9);
10    for i ← 0 to (QUANTIDADE DE PARTÍCULAS-1) do
11        for j ← 0 to (QUANTIDADE DE VARIÁVEIS-1) do
12            Atualizar valor de velocidade da variável j (Equação 1);
13            Aplicar ajuste de velocidade da variável j;
14            Atualizar valor de posição da variável j (Equação 2);
15            Aplicar condições de limites aos valores de posição;
16            Aplicar condições de limites aos valores de velocidade;
17        Avaliar partícula (Fitness);
18        Obter o melhor local da partícula ( $P_{[i][j]}(t)$ )
19        Obter o melhor global do enxame ( $P_{gj}(t)$ )

```

2.2.5 Otimização Baseada em Modelo Sequencial (SMBO)

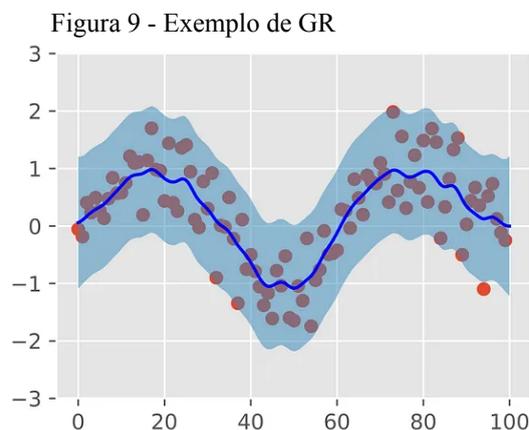
Sequential Model-Based Optimization (SMBO) (BO *et al*, 2022) também chamada de Otimização Bayesiana (BERGSTRA *et al*, 2011), uma das técnicas de otimização utilizadas nesta pesquisa, emprega modelos probabilísticos para encontrar o máximo ou mínimo de uma função, especialmente quando a função é desconhecida, custosa de ser avaliada ou possui muitos parâmetros. Ela envolve a construção de um modelo probabilístico, coleta de dados, atualização do modelo, seleção inteligente de pontos para avaliação e iterações repetidas para encontrar a melhor solução de forma eficaz. É útil em problemas complexos, como otimização de hiperparâmetros em aprendizado de máquina e é especialmente adequada para funções ruidosas ou de alta dimensionalidade.

2.2.5.1 Gaussian Regression (GR)

A GR, também conhecida como Processo Gaussiano (GP), é uma técnica de modelagem estatística que trata a relação entre variáveis como uma distribuição de probabilidade Gaussiana. Neste trabalho, envolveu especificar uma função de média e um kernel de covariância, a Matérn (HANDCOCK *et al*, 2023) com a equação 4 abaixo.

$$K_{Matern}(x, x') = \exp\left(\frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu|d|}}{\ell}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu|d|}}{\ell}\right)\right) \quad (4)$$

Ajustando o modelo com dados do treinamento e usando a inferência bayesiana para fazer previsões em novos pontos, fornecendo distribuições de probabilidade para as previsões. É útil para modelar relações complexas, lidar com dados de alta variância e quantificar a incerteza nas previsões em diversas aplicações, como aprendizado de máquina e previsão em séries temporais.



Fonte:

https://scikit-optimize.github.io/stable/auto_examples/sampler/initial-sampling-method.html#random-sampling
Acessada em: 22, nov 2023.

Na Figura 9, a média prevista é a linha azul, e o intervalo de confiança é a faixa azul claro, para o modelo GPR.

A partir da capacidade de delimitar o espaço amostral com o intervalo de confiança foi possível reduzir os limites da busca de hiperparâmetros no trabalho.

2.2.5.2 *Tree-structured Parzen Estimator* (TPE)

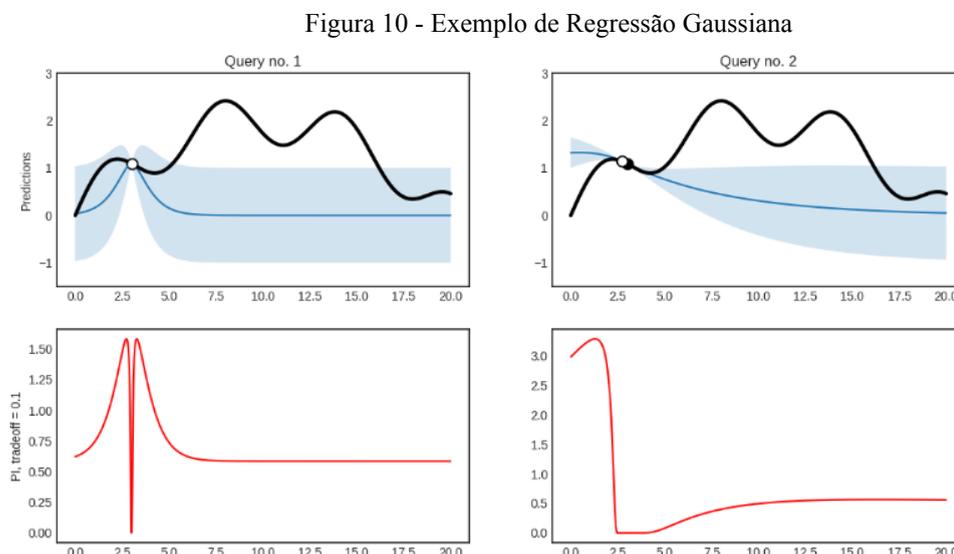
O *Tree-structured Parzen Estimator* (TPE) (NGUYEN *et al*, 2020) é um método de otimização bayesiana utilizado para encontrar a melhor configuração de hiperparâmetros em algoritmos de aprendizado de máquina. Ele constrói uma árvore de probabilidades, onde a cada iteração, diferentes configurações de hiperparâmetros são amostradas com base em modelos estatísticos, e suas performances são avaliadas. Com base nesses resultados, o TPE ajusta os modelos estatísticos para focar na exploração de configurações promissoras, refinando gradualmente a busca em direção a configurações melhores. Isso o torna eficaz para otimizar algoritmos de aprendizado de máquina em espaços de hiperparâmetros complexos.

2.2.5.2 *Surrogate Function* (SF)

A função de aquisição *Expected Improvement* (EI), em português melhora esperada, foi utilizada como uma função substituta (*Surrogate Function*). É uma ferramenta na otimização bayesiana que ajuda a encontrar a melhor solução matematicamente, em determinado espaço (BARDENET; KEGL, 2010). Foi utilizada para estimar o quanto a avaliação adicional da função objetivo em um ponto específico do espaço de entrada pode melhorar a solução atual. O EI utiliza um modelo probabilístico para a função objetivo, calculando a melhoria esperada em relação à melhor solução atual e seleciona o próximo ponto de avaliação com base nesse cálculo. Isso torna a otimização mais eficaz ao equilibrar a exploração de novas áreas com a exploração de melhorias potenciais. O processo é iterativo e continua até um critério de parada ser atingido. Em resumo, o EI é uma técnica que guia a busca pela solução ideal em problemas de otimização. Calculada com a equação 5 abaixo (FRANGIONI, 1995).

$$EI_{n+1} = E[\max(f(x_{n+1}) - f_n, 0)] \quad (5)$$

Na Figura 10, abaixo, pode-se observar o ponto de EI no pequeno círculo branco nas curvas dos gráficos da parte superior da Figura. Nos gráficos da parte inferior da Figura expressa-se as mudanças realizadas com base no ponto de EI.



Fonte:

https://scikit-optimize.github.io/stable/auto_examples/sampler/initial-sampling-method.html#random-sampling
Acessada em: 22, nov 2023.

2.3 Resultados

Os itens abaixo, ilustrados pelas respectivas tabelas, mostram os resultados obtidos no processo de treinamento e avaliação dos modelos de aprendizado de máquina com diferentes técnicas de amostragem de hiperparâmetros.

2.3.1 Fully Connected ANN

A Tabela 1 a seguir apresenta os resultados obtidos para a ANN com os melhores desempenhos assinalados em negrito, comparando as diversas configurações do método proposto pelo trabalho.

Tabela 1 - Estatística dos Resultados Fully Connected ANN

Modelos	Média	Desvio Padrão	Variância	Amostras
SMBO + PSO 50	1.06364733800292	1.00811267681950 03	1.01629116916417 83	SRS
SMBO + PSO 100	0.73490459658205 51	0.89454936755217 64	0.80021857098799 89	
SMBO + PSO 200	1.03482366614043 7	0.96733575050252 11	0.93573845420027 57	

SMBO + PSO 50	1.05292429402470 59	1.00723110673367 74	1.01451450237194 86	LHS
SMBO + PSO 100	1.55953912772238 25	0.90582210302714 2	0.82051368233251 41	
SMBO + PSO 200	1.16479129754006 87	1.05633406298433 36	1.11584165262099 02	
SMBO + PSO 50	1.10593064166605 48	1.02909874994292 8	1.05904423713409 73	Grid Search
SMBO + PSO 100	0.16174013391137 118	0.25414506988280 117	0.06458971654573 388	
SMBO + PSO 200	0.20476258471608 16	0.31413877592843 825	0.09868317054181 754	

Fonte: Elaborada pelo autor (2023)

A primeira coluna apresenta os modelos de treinamento com a técnica PSO e GR, para 50, 100 e 200 partículas, utilizando diferentes técnicas de amostragem, SRS, LHS e GS. Observou-se que em uma ANN simples, os melhores resultados, destacados em negrito, foram obtidos na técnica *Grid Search*. Contudo, na técnica SRS também foi possível encontrar valores que se aproximam dos melhores resultados. Também foi perceptível não haver influência da quantidade de partículas, embora fosse possível supor que a maior quantidade traria melhores resultados.

O melhor resultado expresso na Tabela 1 foi comparado com métodos comuns com demonstrado na Tabela 2 abaixo:

Tabela 2 - Estatística dos Resultados Fully Connected ANN

Modelos	Média	Desvio Padrão	Variância
SMBO + PSO 100(Grid Search)	0.161740133911 37118	0.2541450698828 0117	0.0645897165457 3388
SRS	0.059891301020 97984	0.0013033288603 97604	1.6986661183453 168e-06
GS	0.059636266157 03101	0.0011856889339 1583	1.4058582480104 574e-06
LHS	0.060270741581 916774	0.0013797984547 740199	1.9038437757967 73e-06
TPE	0.078043435301 09946	0.0778981851297 4488	0.0060681272465 08006

Fonte: Elaborada pelo autor (2023)

A comparação expressa na Tabela 2 constatou que o GS obteve os melhores resultados dentre as técnicas experimentadas. Deduziu-se então que a influência do método GS tenha sido responsável pelo alcance de melhores resultados entre os modelos da Tabela 1.

Com os dados da Tabela 2, construiu-se os gráficos constantes da Figura 11, abaixo:

Figura 11 - Gráficos Estatísticos dos Resultados *Fully Connected* ANN



Fonte: Elaborada pelo autor (2023)

Ao se observar os gráficos da Figura 11, considerando média, desvio padrão e variância, o modelo LHS parece apresentar o melhor desempenho geral em termos de consistência e previsibilidade dos resultados, seguido por SRS e GS. O modelo SMBO + PSO 100 (*Grid Search*) tem a maior média, mas sua alta variabilidade o torna menos previsível. TPE, tem a maior média, possui resultados altamente variáveis e menos consistentes. Ressalta-se que a escolha do modelo dependerá dos requisitos específicos do problema e das prioridades em relação à consistência e variabilidade dos resultados.

Somente a análise estatística, apesar de ser uma referência, pode não ser suficiente para evidenciar o melhor desempenho, pois é necessário levar em conta outros fatores numa avaliação mais completa.

2.3.2 *Convolutional Neural Network* (CNN)

Outra técnica empregada foi a CNN e os resultados estão expressos Tabela 3, abaixo.

Tabela 3 - Estatística dos Resultados CNN

Modelos	Média	Desvio Padrão	Variância	Amostras
SMBO + PSO 50	0.09701231718063 352	0.13611522362331 59	0.01852735410202 5294	SRS
SMBO + PSO 100	0.05618950929492 707	0.01681747783836 6686	0.00028282756084 39546	
SMBO + PSO 200	0.11046197246760 128	0.16010607303363 39	0.02563395462225 131	
SMBO + PSO 50	0.13612932637333 866	0.16439311028517 317	0.02702509470923 3106	LHS
SMBO + PSO 100	0.11390486620366 572	0.15614547466472 004	0.02438140925827 0724	
SMBO + PSO 200	0.15774541664868 59	0.20696093800839 926	0.04283282986131 648	
SMBO + PSO 50	0.13244825080037 115	0.17226736916028 13	0.02967604647740 4634	Grid Search
SMBO + PSO 100	0.10805158503353 593	0.14802398662879 784	0.02191110061748 252	
SMBO + PSO 200	0.17932370882481 335	0.21917373068144 166	0.04803712422082 1126	

Fonte: Elaborada pelo autor (2023)

Antes de comentar estes resultados vale a pena registrar que a CNN possui um modelo mais complexo que o da *Fully Connected* ANN. E os números da Tabela 3 indicaram que o método com SRS tornou-se mais eficaz com a arquitetura mais complexa da CNN.

Por outro lado, na Tabela 4 abaixo, onde o melhor resultado da Tabela 3 é comparado com métodos comuns, observou-se um desempenho superior na média na técnica LHS, enquanto em desvio padrão e variância, coube à técnica SRS obter o melhor resultado.

Tabela 4 - Estatística dos Resultados CNN

Modelos	Média	Desvio Padrão	Variância
SMBO + PSO 100(SRS)	0.056189509294 92707	0.0168174778383 66686	0.0002828275608 439546
SRS	0.026644943282 008126	0.0013383995641 576046	1.7913133933372 66e-06
GS	0.026772191654 890733	0.0014556514782 407513	2.1189212261044 843e-06

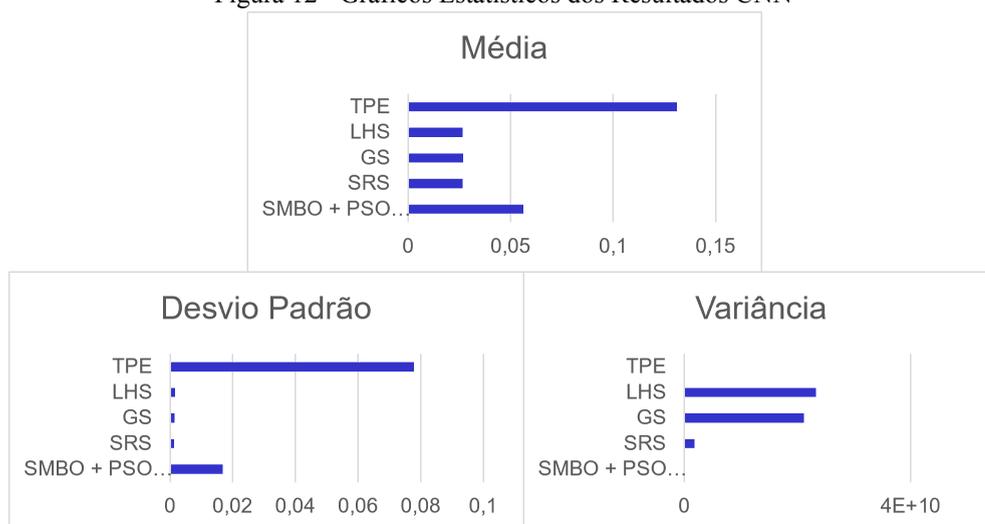
Modelos	Média	Desvio Padrão	Variância
SMBO + PSO 100(SRS)	0.056189509294 92707	0.0168174778383 66686	0.0002828275608 439546
LHS	0.026571318227 797708	0.0015272665547 250328	2.3325431291816 714e-06
TPE	0.131212419165 03223	0.0778981851297 4488	0.0060681272465 08006

Fonte: Elaborada pelo autor (2023)

Apesar da LHS obter a melhor média de resultados, houve uma diferença pequena a seu favor, enquanto a técnica SRS obteve resultados menos erráticos. Pelo comportamento menos estocástico observado, seria possível escolher o SRS, dependendo do problema.

Com os dados da Tabela 4, construiu-se os gráficos constantes da Figura 12 abaixo:

Figura 12 - Gráficos Estatísticos dos Resultados CNN



Fonte: Elaborada pelo autor (2023)

A Figura 12 acima permite observar a distância entre os diversos resultados obtidos. Analisando-se os três resultados juntos (média, desvio padrão e variância) para todos os modelos de otimização de hiperparâmetros em redes neurais, buscando a tendência estatística de melhor desempenho geral no CNN, apenas considerando as referências estatísticas, é possível chegar às seguintes afirmações:

- A técnica SMBO + PSO 100(SRS) possui a segunda maior média entre todas as abordagens, o que indica um desempenho médio ruim na otimização de hiperparâmetros. Também possui o menor desvio padrão e variância, indicando alta consistência e previsibilidade em seus resultados.
- Em contraste, SRS, GS e LHS têm médias significativamente menores e altas variâncias, tornando seus resultados menos confiáveis e imprevisíveis.

- A técnica TPE possui a maior média, mas também o maior desvio padrão e variância, o que sugere que seus resultados são mais variáveis e menos consistentes do que SMBO + PSO 100(SRS).

No geral, considerando média, desvio padrão e variância, a técnica SMBO + PSO 100(SRS) obteve bons resultados, comparando a eficácia em relação ao outro método SMBO, este sendo o TPE. SRS, GS e LHS que apresentam desempenho superior apesar da alta variabilidade em seus resultados.

Contudo, a escolha da técnica somente por essa análise estatística, apesar de ser uma referência, pode não ser o melhor caminho, pois é necessário levar em conta os requisitos específicos do problema e a importância da consistência e previsibilidade dos resultados.

2.4 Discussão

Neste item registra-se as limitações e interpretações dos resultados sobre os fatores que impactam o desempenho das redes neurais, para melhor compreensão desse campo de estudo.

A falta de dados relacionados a treinamentos de ANNs com seus hiperparâmetros para estudos e análise, tornou necessária a geração própria de amostras. Para pesquisas futuras é importante a aquisição ou geração antecipada de amostras.

A necessidade de otimizar variáveis categóricas prejudica a precisão de técnicas como PSO, além de uma dificuldade na generalização dos cálculos por tratar-se de hiperparâmetros contínuos e em intervalos diferentes. O modelo adaptado para problemas com diferentes tipos de variáveis já existe e seria uma possível solução (WANG *et al*, 2021).

A adoção da Loss como métrica principal, simplificou o trabalho mas, também, pode ter sido um fator limitante em relação a conclusões mais amplas. Idem quanto à interpretação das métricas estatísticas em relação ao desempenho das diferentes técnicas. Por exemplo, a diferença de desempenho da média deu uma vantagem para a técnica LHS, mas as demais médias dos demais métodos, foram muito próximas e isso pode indicar a oportunidade de inserir mais elementos de métrica estatística para uma análise mais fina, que permita evidenciar com maior precisão as vantagens de uma técnica sobre a outra.

Instiga-se a pesquisa em futuros trabalhos acerca de afinação automática (*Autotuning*) de hiperparâmetros, pois é possível visualizar a otimização sem a realização de constantes

treinamentos pelo método de tentativa e erro. No caso de uma investigação que aborde a eficiência.

É incentivado o prosseguimento dos estudos na técnica de Inferência Bayesiana, explorando novas funções de aquisição e funções de interpolações no processo de GR.

Apesar das limitações explicitadas, verificou-se uma tendência indicativa de que as metodologias, se integradas como foram nesta pesquisa possuem uma eficácia superior em comparação às técnicas tradicionalmente adotadas, justificando assim a continuidade desta investigação. A metodologia empregada iniciou com análises comparativas intra-metodológicas, ou seja, entre os resultados gerados pelas técnicas selecionadas para este estudo, seguida por comparações inter-metodológicas com abordagens convencionais. Ademais, a adoção da estratégia de validação cruzada revelou uma contribuição prática positiva e relevante para o presente estudo.

2.5 Conclusão/Considerações Finais

Na origem desta pesquisa determinou-se os objetivos a alcançar e, em razão das limitações iniciais, pretendeu-se explorar o potencial da proposta de combinação e comparação de técnicas de otimização. Se, ao final, ocorresse um achado positivo ou negativo, haveria o registro induzindo ou não à continuidade do direcionamento adotado.

Assim, o objetivo geral proposto foi o de investigar a eficácia dos novos métodos na definição de hiperparâmetros em redes neurais por meio de avaliações práticas que seguiram o fluxo no item 2.2.1, Processo de Treinamento e Avaliação de Modelos de ML.

Concluiu-se com êxito o objetivo geral, considerando que se chegou a explorar as técnicas de otimização propostas, experimentá-las em treinamentos e compará-las com outros resultados de otimização que, ao final, permitiu extrair resultados de perdas (*Loss*) indicativos da eficácia de cada modelo.

Ao analisar o primeiro dos objetivos específicos houve a identificação, a revisão e a descrição de cada uma das técnicas definidas para serem empregadas, nomeadamente LHS, SRS, GS, TPE e PSO.

Quanto ao segundo objetivo específico, de realizar simulações utilizando diferentes combinações dos métodos de otimização em diversas arquiteturas de ANN, também conseguiu-se efetivar, com a estratégia de validação cruzada.

E em relação ao terceiro e último objetivo específico, de analisar e comparar os resultados das simulações em diferentes contextos e condições para avaliar a eficácia e adequação dos métodos, idem, o conteúdo de avaliação contido no item 2.3, Resultados, expõe o que se conseguiu a respeito, que aponta para um caminho de sequência em estudos futuros.

Foi observada uma melhora na eficácia na busca de hiperparâmetros utilizando a combinação da Inferência Bayesiana com PSO, em relação ao outro método de Inferência Bayesiana, o TPE.

Estes objetivos caracterizaram o trabalho investigativo, e o desfecho mostrou ser oportuno prosseguir na estratégia de combinação de técnicas até assegurá-la como método mais eficaz e de menor custo na otimização de hiperparâmetros.

A despeito das suas limitações e da necessidade de mais estudos, aumento de banco de dados e comparações com outros métodos de otimização, como gradient-free model-based optimization, esta pesquisa, oferece alento a sua continuidade e traz contribuições específicas para a otimização de hiperparâmetros em ANNs, destacando a importância de abordagens mais precisas na definição desses parâmetros. Além disso, identifica-se direções futuras promissoras para aprimorar ainda mais a eficácia das redes neurais em aplicações diversas.

3 REFERÊNCIAS BIBLIOGRÁFICAS

ABADI, M., et al. (2016). TensorFlow: A system for large-scale machine learning. 12th **USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)**. Disponível em: <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>. Acesso em: 31, mai 2023.

BARDENET, R mi; KEGL, Balazs. Surrogating the surrogate: accelerating Gaussian-process-based global optimization with a mixture cross-entropy algorithm. **ICML 2010**, ICML, jun. 2010. Disponível em <<https://icml.cc/Conferences/2010/papers/297.pdf>>. Acesso em: 31 mai. 2023.

B. A. S. Emambocus. et al. A Survey on the Optimization of Artificial Neural Networks Using Swarm Intelligence Algorithms. **IEEE Access**, vol. 11, p.1280-1294, 2023, jan. 2023.

BERGSTRA, J., BARDENET, R., BENGIO, Y., & KÉGL, B., **Algorithms for hyper-parameter optimization**, **Advances in neural information processing systems** **24**, 2011. Disponível em <https://papers.nips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html>. Acesso em: 02 jun. 2023.

BERGSTRA, J.; BENGIO, Yoshua. Random search for hyper-parameter optimization. **Journal of Machine Learning research**, v. 13, n. 2, 2012.

BERGSTRA, J.; YAMINS, D.; COX, David. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: **International conference on Machine Learning**. PMLR, 2013. p. 115-123.

BERGSTRA, J., YAMINS, D.; COX, David.. (2013). Hyperopt: A Python library for optimizing the hyperparameters of machine learning algorithms. **Proceedings of the 12th Python in Science Conference**. Disponível em: https://conference.scipy.org/proceedings/scipy2013/pdfs/bergstra_hyperopt.pdf. Acesso em: 31, mai 2023.

BO, Yin et al. **Real-time hard-rock tunnel prediction model for rock mass classification using CatBoost integrated with Sequential Model-Based Optimization**. **Tunnelling and underground space technology**, v. 124, p. 104448, 2022.

BYERLY, Adam; KALGANOVA, Tatiana; DEAR, Ian. No routing needed between capsules. **Neurocomputing**, v. 463, p. 545-553, 2021.

CHONGZHI, Wu; LIN, Wang; ZHANG, Wengang. **Assessment of undrained shear strength using ensemble learning based on Bayesian hyperparameter optimization**. In: **Modeling in Geotechnical Engineering**. Academic Press, 2021. p. 309-326.

CLAESEN, M.; DE MOOR, B. **Hyperparameter Search in Machine Learning**. [s.l: s.n.]. Disponível em: <<https://browse.arxiv.org/pdf/1502.02127.pdf>>. Acesso em: 9 out. 2023.

DANIEL, Leal. **Intercomunicação dinâmica e pesquisa baseada em cruzamento BLX com histórico de trajetória aplicada ao algoritmo de otimização por multi-enxame de partículas com parâmetros de controle semi-estocásticos**. 180. 2021. Tese (Programa de Pós-Graduação em Engenharia Elétrica) - Instituto de Tecnologia, Universidade Federal do Pará, Pará, Belém, 2021

FRANGIONI, Antonio. On a new class of bilevel programming problems and its use for reformulating mixed integer problems. **European journal of operational research**, v. 82, n. 3, p. 615-646, 1995.

GOODFELLOW, I., Bengio, Y., & Courville, A. (2016). **Deep Learning**. MIT Press. Disponível em: <https://www.nature.com/articles/nature14539>. Acesso em: 31, mai 2023.

GUTMANN, H.-M. A radial basis function method for global optimization. **Journal of Global Optimization** 19, SpringerLink, Issue 3, p. 201–227, mar. 2001. Disponível em <<https://doi.org/10.1023/A:1011255519438>>. Acesso em: 02 jun. 2023.

HANDCOCK, Mark S., and Michael L. Stein. “A Bayesian Analysis of Kriging.” *Technometrics*, vol. 35, no. 4, 1993, pp. 403–10. JSTOR, <https://doi.org/10.2307/1270273>. Acesso em: 8, Out. 2023.

HARRIS, C. R., et al. (2020). Array programming with NumPy. **Nature**, 585(7825), 357-362. Disponível em: <https://www.nature.com/articles/s41586-020-2649-2>. Acesso em: 31, mai 2023.

HIRATA, Daiki; TAKAHASHI, Norikazu. **Ensemble learning in CNN augmented with fully connected subnetworks**. IEICE TRANSACTIONS on Information and Systems, v. 106, n. 7, p. 1258-1261, 2023.

JADERBERG, Max et al. Population based training of neural networks. **arXiv preprint arXiv:1711.09846**, 2017.

KENNEDY, J; EBERHART, R. **Particle swarm optimization**, Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. **Nature**, 521(7553), 436-444. Disponível em: <https://www.nature.com/articles/nature14539>. Acesso em: 31, mai 2023.

LI, Y.; ZHANG, Y.; ZHOU, G. et al., "**Bayesian Optimization with Particle Swarm**," 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 2021, pp. 1-6, doi: 10.1109/IJCNN52387.2021.9533761.

MCKINNEY, W. (2010). Data Structures for Statistical Computing in Python. **Proceedings of the 9th Python in Science Conference**. Disponível em <https://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf>. Acesso em: 31, mai 2023.

NGUYEN, Hoang-Phuong; LIU, Jie; ZIO, Enrico. A long-term prediction approach based on long short-term memory neural networks with automatic parameter optimization by Tree-structured Parzen Estimator and applied to time-series data of NPP steam generators. *Applied Soft Computing*, v. 89, p. 106116, 2020.

PANDA, Balaram. (2019). **Hyperparameter Tuning**. 10.13140/RG.2.2.11820.21128.

PEDREGOSA, F.. Hyperparameter optimization with approximate gradient. In: **International Conference on Machine Learning**. PMLR, 2016. p. 737-746.

PEDREGOSA, F., et al. (2011). Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, 12, 2825-2830. Disponível em: <https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>. Acesso em: 31, mai 2023.

POLI, R., Kennedy, J. & Blackwell, T. Particle swarm optimization. *Swarm Intell* 1, 33–57 (2007). <https://doi.org/10.1007/s11721-007-0002-0>

SHAMBHU, Rai. Influence of Artificial Technology in Recent Times Disponível em: <<https://www.linkedin.com/pulse/influence-artificial-technology-recent-times-shambhu-rai/>>. Acesso em: 11 out. 2023.

SCIENCE DIRECT. Latin Hypercube. Disponível em <<https://www.sciencedirect.com/topics/mathematics/latin-hypercube>>. Acesso em: 2, jun. 2023.

SCIKIT-LEARN. Gaussian Processes. Disponível em <https://scikit-learn.org/stable/modules/gaussian_process.html>. Acesso em: 18 set. 2023.

SHAZEER, Noam; STERN, Mitchell. Adafactor: Adaptive learning rates with sublinear memory cost. In: *International Conference on Machine Learning*. PMLR, 2018. p. 4596-4604.

SHI, Yuhui et al. Particle swarm optimization: developments, applications and resources. In: *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*. IEEE, 2001. p. 81-86.

SHI, Yuhui; EBERHART, Russell. A modified particle swarm optimizer. In: **1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)**. IEEE, 1998. p. 69-73.

SINGH, S. (2003). Simple Random Sampling. In: *Advanced Sampling Theory with Applications*. Springer, Dordrecht. Disponível em: https://doi.org/10.1007/978-94-007-0789-4_2. Acesso em: 2, jun. 2023.

TOWARDS DATA SCIENCE. **Building a Tree-Structured Parzen Estimator from Scratch (Kind Of)**. Disponível em <<https://towardsdatascience.com/building-a-tree-structured-parzen-estimator-from-scratch-kind-of-20ed31770478>>. Acesso em: 2, jun. 2023.

TOWARDS DATA SCIENCE. **Machine Learning for Beginners: An Introduction to Neural Networks**. Disponível em: <<https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-d49f22d238f9>>. Acesso em: 22, nov. 2023.

VIRTANEN, P., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3), 261-272. Disponível em <https://www.nature.com/articles/s41592-019-0686-2>. Acesso em: 31, mai 2023.

WANG, Feng; ZHANG, Heng; ZHOU, Aimin. A particle swarm optimization algorithm for mixed-variable optimization problems. *Swarm and Evolutionary Computation*, v. 60, p. 100808, 2021.

WANG, J.; Yan, Z.; Pan, T.; Zhu, Z.; Song, X.; Yang, D. Drilling Parameters Multi-Objective Optimization Method Based on PSO-Bi-LSTM. *Appl. Sci.* 2023, 13, 11666. Disponível em <https://doi.org/10.3390/app132111666>, Acesso em: 30, nov. 2023.

WATANABE, Shuhei. Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance.**arXiv preprint arXiv:2304.11127**,arXiv, mai. 2023.Disponível em <<https://arxiv.org/abs/2304.11127v3>>. Acesso em: 02, jun e 2023.