Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas

AutoML Multiobjetivo, uma abordagem sob a perspectiva da Fronteira de Pareto.

Jean Pieri Angelo de Matos

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:
Prof. Talles Henrique de Medeiros

Fevereiro, 2024
João Monlevade–MG

Jean Pieri Angelo de Matos

AutoML Multiobjetivo, uma abordagem sob a perspectiva da Fronteira de Pareto.

Orientador: Prof. Talles Henrique de Medeiros

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina "Trabalho de Conclusão de Curso II".

Universidade Federal de Ouro Preto

João Monlevade

Fevereiro de 2024

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

M433a Matos, Jean Pieri Angelo de.
AutoML multiobjetivo, uma abordagem sob a perspectiva da Fronteira de Pareto. [manuscrito] / Jean Pieri Angelo de Matos. - 2024.
39 f.: il.: color., gráf., tab.. + Algoritmo.

Orientador: Prof. Dr. Talles Henrique de Medeiros.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia de Computação .

1. Algoritmos genéticos. 2. Inteligência artificial. 3. Redes neurais (Computação). 4. Sistemas de Computação. I. Medeiros, Talles Henrique de. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.8

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTACAO E SISTEMAS

FOLHA DE APROVAÇÃO

Jean Pieri Angelo de Matos

AutoML Multiobjetivo, uma Abordagem sob a Perspectiva da Fronteira de Pareto.

Monografia apresentada ao Curso de Engenharia da Computação da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação

Aprovada em 21 de fevereiro de 2024.

Membros da banca

Doutor - Talles Henrique de Medeiros - Orientador - Universidade Federal de Ouro Preto
Doutor - Eduardo da Silva Ribeiro - Universidade Federal de Ouro Preto
Doutor - Luiz Carlos Bambirra Torres - Universidade Federal de Ouro Preto

Talles Henrique de Medeiros, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 02/03/2024.

Documento assinado eletronicamente por **Talles Henrique de Medeiros**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 04/03/2024, às 21:08, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0671133** e o código CRC **2EF34DFF**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3808-0819 - www.ufop.br

"Science is more than a body of knowledge; it is a way of thinking."

— Carl Sagan (1934 – 1996),
in: The Demon-Haunted World: Science as a Candle in the Dark.

Resumo

Este trabalho consiste no desenvolvimento de um algoritmo genético direcionado para a otimização de Redes Neurais Convolucionais (CNN) utilizando uma abordagem multiobjetivo. O intuito é alcançar um modelo final otimizado sem a necessidade de intervenção humana, visando democratizar o uso da IA para o público não especializado e torná-lo acessível também a indivíduos sem especialização específica. Os resultados obtidos em três experimentos distintos confirmaram a eficácia do algoritmo proposto. Através da análise criteriosa dos resultados, foi possível identificar e selecionar o melhor modelo conforme os objetivos estabelecidos. Essa abordagem permitiu não apenas a otimização dos modelos de Redes Neurais Convolucionais (CNN), mas também a validação da metodologia proposta. Esses resultados reforçam a viabilidade da utilização de um algoritmo genético em conjunto com a Fronteira de Pareto em conjunto e a Abordagem Lexicográfica e demonstra a capacidade de encontrar soluções ótimas em um espaço de busca complexo e multidimensional, reduzindo o espaço de busca na fronteira e eliminando soluções onde um objetivo prevalece de maneira significativa sobre o outro. Código disponível em <https://github.com/jpamatos/multiobjective>.

Palavras-chave: AutoML. Fronteira de Pareto. Abordagem Lexicográfica. Algoritmo Genético.

Abstract

This work consists of the development of a genetic algorithm aimed at optimizing Convolutional Neural Networks (CNNs) using a multi-objective approach. The aim is to achieve an optimized final model without the need for human intervention, aiming to democratize the use of AI for the non-specialized public and also make it accessible to individuals without specific specialization. This approach allowed not only the optimization of CNNs models, but also the validation of the proposed methodology. These results reinforce the feasibility of using a genetic algorithm in conjunction with the Pareto Frontier and the Lexicographic Approach and demonstrate the ability to find optimal solutions in a complex and multidimensional search space, reducing the search space at the frontier and eliminating solutions where one objective significantly prevails over the other. Code available in <https://github.com/jpamatos/multiobjective>.

Key-words: AutoML. Pareto frontier. Lexicographic Approach. Genetic Algorithm.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AutoML Automated Machine Learning

NAS Neural Architecture Search

DL Deep Learning

IA Inteligência Artificial

AG Algoritmos Genéticos

CNN Redes Neurais Convolucionais

SVM Máquina de Vetores de Suporte

KNN K-Vizinhos Mais Próximos

RNN Redes Neurais Recorrentes

Lista de símbolos

n_2	Representação binária
n_{10}	Representação decimal
$\leq$	Menor ou igual a
$>$	Maior que
$\forall$	Para todo
$\mid$	Tal que
$\nexists$	Não existe
$\wedge$	e (conjunção)

Sumário

1 INTRODUÇÃO

Os progressos na área de Aprendizado de Máquina e Inteligência Artificial (IA) refletem frequentemente a crescente complexidade dos problemas a serem abordados. Isso se traduz em algoritmos e modelos mais sofisticados, proporcionando ferramentas poderosas para uma ampla gama de aplicações.

Conforme os estudos de Jentzen, Kuckuck e Wurstemberger (2023) as Redes Neurais são modelos matemáticos capazes de aprender padrões nos dados fornecidos. Frequentemente, esses modelos funcionam como caixas-pretas, o que significa que os usuários podem não compreender completamente seu funcionamento interno. Isso pode aumentar a distância percebida entre o usuário comum e os modelos de IA.

O Deep Learning tem demonstrado resultados notáveis em uma variedade de tarefas desafiadoras, como reconhecimento de padrões em imagens, tradução automática, processamento de linguagem natural, entre outras, conforme mencionado por Chagas (2019). Sua flexibilidade e capacidade de escalabilidade tornaram-no uma ferramenta indispensável em diversos domínios, incluindo visão computacional, biologia computacional, medicina e finanças.

Com o avanço do Aprendizado Profundo, o Automated Machine Learning (AutoML) surgiu como uma resposta à crescente complexidade associada à concepção, treinamento e otimização de modelos de aprendizado de máquina. De acordo com (HE; ZHAO; CHU, 2021), o AutoML visa automatizar todo o processo de desenvolvimento de modelos, desde a seleção e pré-processamento de dados até a escolha dos algoritmos de aprendizado e ajuste dos hiperparâmetros. Ao capacitar até mesmo usuários sem um conhecimento técnico profundo em aprendizado de máquina a criarem e implantarem modelos de alta qualidade, o AutoML tem democratizado o acesso à IA e acelerado o desenvolvimento de soluções inteligentes em uma variedade de domínios, conforme observado por (TRUSS; SCHMITT, 2024).

1.1 Elaboração do capítulo

Este trabalho de conclusão de curso deverá permitir que seja possível o estudo, a análise e a implementação de métodos para se testar algoritmos de Neural Architecture Search (NAS), sob a perspectiva multiobjetivo, para obter um modelo final de rede neural em um problema de aprendizagem profunda.

1.2 O Problema de pesquisa

Permitir que as redes neurais profundas possam ser treinadas com uma menor interferência humana no projeto da arquitetura e de seus hiperparâmetros. Esse esforço visa democratizar o acesso a essas ferramentas de Inteligência Artificial (IA), tornando os resultados mais acessíveis e aproximando a IA do público não especializado. Além disso, há uma investigação significativa sobre a possibilidade de viés por parte de especialistas nas soluções encontradas, o que pode restringir o espaço de busca dos algoritmos e limitar a eficácia dos resultados obtidos. Introduzir a AutoML como protagonista no processo de aprendizado pode proporcionar um tratamento mais equitativo a diversos aspectos cruciais do processo completo de aprendizado de máquina. Este trabalho propõe o desenvolvimento de um algoritmo baseado em Algoritmos Genéticos (AG) para explorar um espaço de busca de soluções e identificar a ótima, empregando a Fronteira de Pareto e a Abordagem Lexicográfica.

1.3 Objetivos

Desenvolver um método de otimização multiobjetivo voltado para a minimização da interferência humana no treinamento de redes neurais profundas. O foco será na criação de um método que reduza a necessidade de intervenção humana e conhecimento especializado durante o processo de treinamento, permitindo uma maior automação e eficiência no desenvolvimento de modelos de aprendizado profundo.

Este trabalho possui como objetivos específicos:

- Desenvolver um algoritmo genético multiobjetivo para treinamento de redes profundas com diferentes objetivos.
- Implementar uma abordagem lexicográfica para direcionar a busca por soluções preferenciais, simplificando o conhecimento do usuário sobre a importância dos objetivos.
- Validar o algoritmo proposto e desenvolvido com o popular dataset MNIST.
- Analisar experimentalmente o impacto dessas métricas em um problema de aprendizagem profunda.

1.4 Metodologia

O estudo visa encontrar a melhor arquitetura de rede neural convolucional para o conjunto de dados MNIST, otimizando simultaneamente dois objetivos à escolha do usuário.

Os passos para execução deste trabalho são assim definidos:

- Revisão da literatura relacionada a otimização multiobjetivo, algoritmos genéticos, redes neurais e abordagens lexicográficas.
- Desenvolvimento de um algoritmo genético personalizado para realizar a otimização multiobjetivo das arquiteturas de rede neural. Isso inclui desenvolver operadores de seleção, cruzamento e mutação.
- Validação do algoritmo genético em um caso de estudo real relacionado ao domínio de interesse.
- Análise dos resultados obtidos com a aplicação da metodologia e a eficácia da abordagem lexicográfica. Discussão das implicações dos resultados e identificar possíveis áreas para futuras investigações.

1.5 Organização do trabalho

O restante deste trabalho é organizado como se segue. O Capítulo 2 apresenta a revisão da literatura, onde foi explorado trabalhos disponíveis no campo do AutoML focado na otimização multiobjetivo. O Capítulo 3 mostra detalhes da implementação do algoritmo utilizado para otimização. O Capítulo 4 traz uma discussão do que foi observado. O Capítulo 5 fecha as discussões apresentadas ao longo do trabalho.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo serão apresentados importantes contribuições na literatura referentes aos tópicos que são abordados neste trabalho de conclusão.

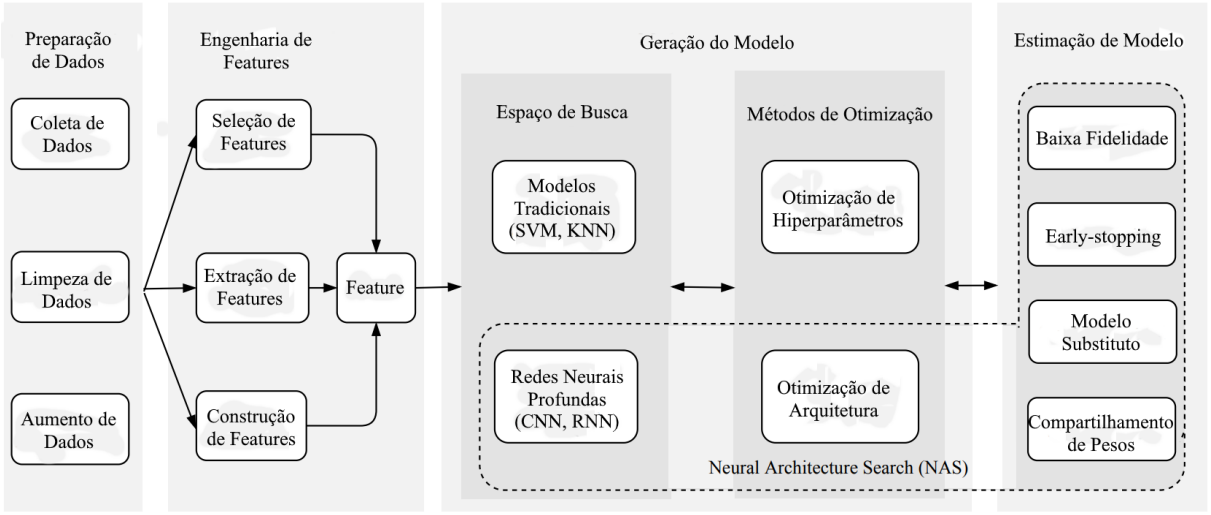
2.1 O AutoML

O Automated Machine Learning (AutoML) representa uma solução promissora para a construção de modelos de inteligência artificial sem a necessidade de intervenção humana. De acordo com Wistuba, Rawat e Pedapati (2019) a busca de arquiteturas de rede neural é uma etapa crítica que requer um alto domínio de conhecimento, tornando a automatização desse processo uma abordagem interessante para tornar o Deep Learning (DL) mais acessível. He, Zhao e Chu (2021) realizam uma revisão do estado da arte, destacando que a Neural Architecture Search (NAS) é crucial para o AutoML. Por outro lado, Hsu et al. (2018) abordam o NAS por meio do Aprendizado por Reforço. O AutoML pode ser usado em diversas etapas do Aprendizado de Máquina, esse fato é exemplificado por He, Zhao e Chu (2021) na Figura 1, onde é percebido a aplicação do AutoML antes mesmo da construção do modelo, na extração, coleta e aumento dos dados, na seleção, extração e construção de features. Na geração do modelo, ele pode ser empregado nos espaços de busca para a utilização de diversos modelos tradicionais de Aprendizado de Máquina, como Máquina de Vetores de Suporte (SVM) e K-Vizinhos Mais Próximos (KNN) e também modelos de Aprendizado Profundo, como Redes Neurais Recorrentes (RNN) e Redes Neurais Convolucionais (CNN). Como também pode ser empregado na Otimização de Hiperparâmetros e Otimização de Arquitetura. Este trabalho envolve os Métodos de Otimização, implementando o AutoML na Otimização de Hiperparâmetros.

Existem várias abordagens que podem ser empregadas no AutoML, em concordância com Wistuba, Rawat e Pedapati (2019). Os autores mencionam métodos mono objetivo, multiobjetivo, aprendizado por reforço e métodos evolucionários como algumas das opções disponíveis.

Em contraste com a otimização unidimensional, onde se busca encontrar uma única solução ótima, a otimização multiobjetivo busca encontrar um conjunto de soluções que representem um compromisso entre os diferentes objetivos. Se tornando fundamental para o sucesso do AutoML, permitindo a exploração eficaz de múltiplos objetivos, o equilíbrio entre *trade-offs* e a geração de conjuntos de soluções eficientes e adaptáveis. Na próxima seção será abordado como a análise multiobjetivo é realizada.

Figura 1 – Diagrama do AutoML



Fonte: He, Zhao e Chu (2021)

2.2 A Otimização Multiobjetivo

A otimização multiobjetivo desempenha um papel fundamental em problemas que envolvem a consideração de múltiplos critérios de desempenho. Quando se trata de redes neurais, esses critérios geralmente correspondem às métricas de desempenho. No trabalho realizado por Guerrero-Viu et al. (2021), as redes neurais são otimizadas em relação à arquitetura e à otimização de hiperparâmetros com base em mais de um objetivo.

Foi observado no trabalho de (QIAN et al., 2021) que, ao analisar mais de um objetivo, podem surgir múltiplas funções objetivo conflitantes. Portanto, durante a otimização, torna-se impossível obter uma única solução ótima. Em vez disso, é necessário considerar múltiplos valores objetivos para obter um conjunto de soluções ótimas de compromisso, também conhecido como fronteira de Pareto. Esse conjunto de soluções representa as melhores soluções possíveis, considerando o trade-off entre os diferentes objetivos.

A otimização multiobjetivo lida com a complexidade de encontrar um conjunto de soluções que representem o melhor compromisso entre os diferentes objetivos. Duas abordagens amplamente utilizadas são as técnicas de escalarização linear dos objetivos e a abordagem Lexicográfica. No trabalho de Freitas (2004), essas duas técnicas são comparadas juntamente com a abordagem baseada na Fronteira de Pareto. A primeira abordagem analisada é a soma ponderada dos objetivos que resulta na conversão de um problema multiobjetivo em um problema mono objetivo, mostrada pela Equação 2.1.

$$Q = w_1 * c_1 + w_2 * c_2 + \ \ldots \ + w_n * c_n \tag{2.1}$$

Onde w representa os pesos e c representa os objetivos. A principal vantagem de se utilizar tal abordagem é a simplicidade, resolvendo o problema de forma trivial. E um

dos principais problema é justamente a conversão, onde o problema a se otimizar não é mais um problema multiobjetivo, torando o problema muito sensível à escolha dos pesos. Adicionalmente, esse método enfrenta dificuldades ao lidar com objetivos que possuem magnitudes e unidades de medida diferentes, e também tende a desconsiderar as relações entre os objetivos.

A tabela 1 mostra outros trabalhos relacionados à otimização multiobjetivo e Redes Neurais.

O emprego da otimização multiobjetivo em problemas de otimização foi observado no estudo realizado por Linczuk e Bastos (2020). Dentro das soluções encontradas de gasto energético e conforto térmico em construções, os pesquisadores decidiram pela melhor utilizando a Fronteira de Pareto Ótima.

A Fronteira de Pareto analisa o conjunto de soluções em relação à dominância entre elas. Podendo as soluções serem classificadas como dominadas, que é quando uma solução é pior que outra em todos os objetivos simultaneamente; e não-dominadas, que é quando uma solução não pode ser considerada pior que outra em todos os objetivos simultaneamente. A este sub-conjunto de soluções não-dominadas, denominamos Conjunto Pareto-ótimo de soluções. Isso permite identificar as soluções que não podem ser melhoradas em um objetivo sem piorar em outro. A Figura 2 ilustra esse conceito.

Nessa figura, cada ponto representa uma solução possível, os pontos na cor azul representam as Soluções Dominadas e os pontos na cor verde as Soluções Não Dominadas. As soluções ao longo da curva de Pareto (ou fronteira de Pareto) são consideradas ótimas e representam compromissos ideais entre os objetivos considerados. Essas soluções são interpretadas como tendo importância equivalente. Observa-se que a melhoria de um objetivo em detrimento do outro é evidente ao longo da fronteira de Pareto. Ao buscar aprimorar o objetivo $F_1(x)$, por exemplo, há uma deterioração no desempenho do objetivo $F_2(x)$.

A fronteira de Pareto é composta por soluções não dominadas. No trabalho de Hashimoto (2004) é explicado que em um problema de minimização, uma solução é não dominada se não for possível melhorar um objetivo sem piorar outro. A Equação 2.2, conforme mencionada pelo autor, é utilizada para determinar se uma solução x domina uma solução y. Essa equação é fundamental para avaliar a relação de dominância entre soluções em problemas de otimização multiobjetivo.

$$f_i(x) \leq f_i(y), \forall i \tag{2.2}$$

Onde i representa os objetivos a serem minimizados. A Fronteira de Pareto é o conjunto onde as soluções são não dominadas em relação ao resto do conjunto.

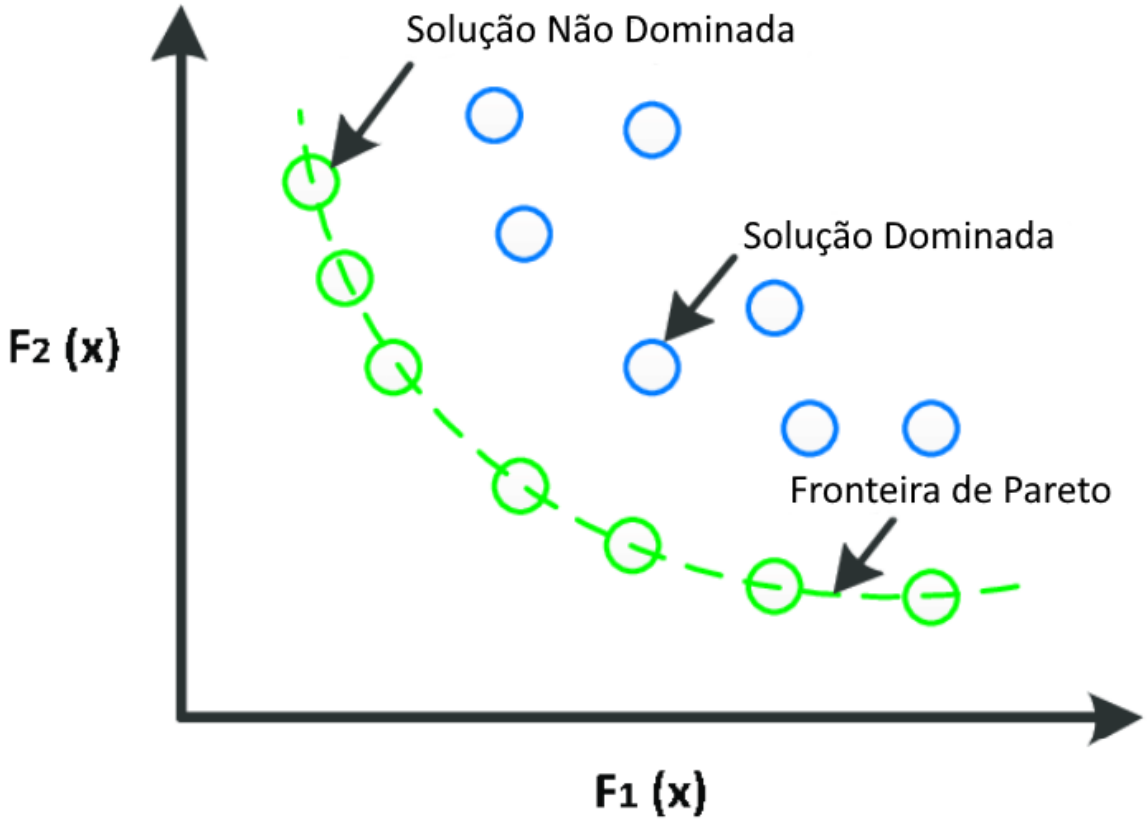
Tabela 1 – Trabalhos que envolvem otimização multiobjetivo em redes neurais

Autor	Contribuição	Tipo de NN
Cong e Zhou (2023)	Classificação de algoritmos de otimização para NAS	CNN
Hu et al. (2021)	Estimar o desempenho da CNN de forma rápida utilizando RWE(Random-Weight Evaluation)	CNN
Elsken, Metzen e Hutter (2019)	Algoritmo multiobjetivo evolucionário para busca de arquitetura, utiliza mecanismo de herança Lamarckian	CNN, DL
Pfisterer et al. (2021)	Otimiza um segundo critério definido pelo usuário levando a um pipeline de ML otimizado	CNN
Wang et al. (2020)	Algoritmo que utiliza os resultados de pesquisa e informações históricas para encontrar arquiteturas que são leves e precisas	CNN
Chen et al. (2020)	Aproxima a eficiência de Pareto rapidamente, as células descobertas alcançam performance de predições comparadas a outros métodos multiobjetivos	RNN
Hu et al. (2020)	Uma nova estratégia para estimar desempenho, RWE para reduzir o custo do NAS	CNN
Falanti et al. (2022)	POPNASv2 nova versão do POPNAS, melhora o desempenho da primeira versão adicionando novos operadores e melhorando a qualidade dos preditores	CNN
Lu et al. (2020a)	NSGANetV2 framework que se equipara ou supera modelos existentes e com ordens de magnitude mais eficientes em se tratando de amostras	CNN
Lu et al. (2023)	EvoXBench, framework genérico que pode ser utilizado sem a necessidade de GPU ou softwares sofisticados como TensorFlow/PyTorch	CNN, DNN

Fonte: Produzido pelo autor

Ao contrário de métodos que requerem a atribuição de pesos aos diferentes objetivos, Freitas (2004) observa que a Fronteira de Pareto não impõe essa necessidade, proporcionando uma análise mais imparcial e flexível. No entanto, sua determinação pode ser computacionalmente intensiva, especialmente em problemas complexos, e a interpretação e seleção das soluções da Fronteira de Pareto podem ser desafiadoras. Apesar dessas

Figura 2 – Fronteira de Pareto para dois objetivos $F_1(x)$ e $F_2(x)$ que buscam ser minimizados.



Fonte: Mahesh, Nallagownden e Elamvazuthi (2016)

limitações, a Fronteira de Pareto continua sendo uma ferramenta valiosa para explorar e identificar um conjunto diversificado de soluções eficientes em problemas multiobjetivo.

Outro método analisado é a Abordagem Lexicográfica que consiste em atribuir diferentes prioridades aos objetivos diferentes e, em seguida, focar na otimização dos objetivos em sua ordem de prioridade. Ela oferece vantagens distintas, incluindo facilidade de interpretação e redução da complexidade. As soluções resultantes são facilmente compreendidas, pois são ordenadas segundo a prioridade atribuída a cada critério, simplificando a análise e a tomada de decisões. Além disso, ao tratar cada critério individualmente, essa abordagem possibilita uma análise mais detalhada da qualidade da solução, permitindo identificar pontos fortes e fracos em cada aspecto específico. No entanto, uma desvantagem significativa é a necessidade de especificar limites de tolerância para cada critério, uma tarefa que pode ser desafiadora e subjetiva, dificultando a definição desses limites de forma consistente e objetiva.

A conclusão apresentada por Freitas (2004) sugere que tanto a abordagem Lexicográfica quanto a abordagem da Fronteira de Pareto deveriam ser mais amplamente adotadas na comunidade de mineração de dados. Isso ressalta a importância de explorar e

aplicar essas abordagens para lidar eficazmente com problemas de otimização multiobjetivo, na prática, sendo a escolha para serem utilizados no desenvolvimento deste trabalho.

2.2.1 A abordagem Lexicográfica

Na abordagem lexicográfica de otimização, os objetivos são ordenados por prioridade e otimizados sequencialmente. O usuário precisa definir uma ordem de prioridade para os objetivos do problema de otimização e, então, ela será seguida para otimizar o modelo em questão. Isso é bastante intuitivo em diversos aspectos do mundo real, especialmente em aprendizado de máquina. No trabalho de Rastegar e Khorram (2015) é mostrado que ao utilizar essa abordagem, o objetivo mais importante é totalmente otimizado antes de passar para o próximo.

O processo de otimização lexicográfica começa otimizando o objetivo mais importante. Isso é feito construindo-se um conjunto de soluções que satisfazem a um limiar pré-definido para esse objetivo. A partir desse conjunto, a melhor solução é selecionada.

Em seguida, passa-se para o próximo objetivo na ordem de prioridade. Novamente, é definido um limite para esse objetivo, e são selecionadas as soluções que se encontram dentro desse limite. Se apenas uma solução estiver dentro do limite, ela é escolhida como a solução ótima para esse objetivo. No entanto, se várias soluções estiverem dentro do limite, é escolhida a melhor delas com base no objetivo anteriormente otimizado, continuando esse processo até que todos os objetivos tenham sido otimizados, O algoritmo 1, exibe esse comportamento em forma de pseudocódigo, para verificar a melhor solução em um conjunto de soluções S, onde t_1 e t_2 são os limiares do primeiro e segundo objetivo respectivamente.

Essa estratégia garante uma exploração eficiente do espaço de soluções, delimitando a área de interesse ao descartar soluções onde um objetivo é otimizado em detrimento do outro. No entanto, é crucial observar que essa abordagem pode resultar em soluções sub ótimas se a ordem de prioridade não estiver claramente definida.

O estudo de Brookhouse e Freitas (2023) se concentra na seleção justa de recursos para classificação, isto é, métodos que escolhem um subconjunto de recursos visando maximizar a precisão e a *fairness* das previsões feitas por um classificador. Mais especificamente, foram comparados dois Algoritmos Genéticos recentemente propostos para a seleção justa de características, os quais são baseados em duas abordagens diferentes de otimização multiobjetivo: (a) um AG baseado na dominância de Pareto; e (b) um AG baseado na otimização lexicográfica, onde maximizar a precisão tem uma prioridade maior do que maximizar a equidade. Os resultados indicam que, em geral, o algoritmo genético lexicográfico superou o algoritmo genético de Pareto em termos de precisão, sem comprometer a equidade dos classificadores aprendidos.

Algoritmo 1: Escolha da melhor solução

Entrada: S, t_1, t_2
Saída: melhor_solucao
$y \leftarrow melhor\_primeiro\_objetivo(S)$
$Q \leftarrow [\,]$
foreach $s \in S$ **do**
  if $|f_1(s) - f_1(y)| < t_1$ **then**
    $Q \leftarrow s$
end
$y \leftarrow melhor\_segundo\_objetivo(Q)$
$R \leftarrow [\,]$
foreach $q \in Q$ **do**
  if $|f_2(q) - f_2(y)| < t_2$ **then**
    $R \leftarrow q$
end
if $tam(R) == 1$ **then**
  $melhor\_solucao \leftarrow R[0]$
else
  $melhor\_solucao \leftarrow melhor\_primeiro\_objetivo(R)$
end

2.3 Os algoritmos genéticos

Os Algoritmos Genéticos (AG) são uma das aplicações evolucionárias mais amplamente utilizadas e bem-sucedidas em conjunto com Redes Neurais Artificiais (RNAs). Baseados em conceitos da biologia evolutiva, como herança, mutação e seleção natural, os AG são inspirados nas teorias de Evolução Natural das Espécies de Charles Darwin [Darwin, 1859]. Silva e Ludermir (2021) retrata que as Redes Neurais utilizadas em conjunto com os Algoritmos Genéticos (AG) conseguem explorar de forma mais eficaz uma ampla variedade de aspectos e componentes necessários, como pesos, arquiteturas e funções de transição, para a construção de RNAs com capacidade de generalização. Elas fazem uso dos AGs para explorar o espaço de soluções e inicializar seus próprios parâmetros, permitindo um melhor aproveitamento (exploitation) e exploração (exploration) desses aspectos.

Os AG reproduzem o processo de seleção natural, no qual os indivíduos mais adaptados têm maior probabilidade de sobrevivência e reprodução, passando suas características para as gerações seguintes. Pacheco (2004) descreve os componentes necessários para a execução de um algoritmo genético, incluindo um diagrama esquemático que destaca as etapas importantes e os operadores clássicos que governam a evolução.

Um diagrama esquemático de um algoritmo genético representa geralmente as seguintes etapas:

- Inicialização: Inicialização da população de soluções candidatas de forma aleatória.

- Avaliação: Avaliação de cada indivíduo da população de acordo com uma função de aptidão.
- Seleção: Seleção dos indivíduos mais aptos para reprodução, com base em sua aptidão.
- Recombinação (Crossover): Cruzamento ou combinação de partes de dois indivíduos para gerar descendentes.
- Mutação: Introdução aleatória de mudanças nos indivíduos para manter diversidade genética.
- Substituição: Substituição dos indivíduos menos aptos da população por novos descendentes.
- Critério de Parada: Critério que determina quando o algoritmo atinge a condição de parada, como número máximo de gerações ou convergência para uma solução satisfatória.

Os operadores clássicos dos algoritmos genéticos incluem:

- Seleção de Pais: Operadores de seleção, como roleta, torneio ou classificação, determinam quais indivíduos serão selecionados para reprodução com base em sua aptidão.
- Crossover (Cruzamento): O crossover combina partes dos cromossomos dos pais para gerar descendentes.
- Mutação: A mutação introduz pequenas alterações aleatórias nos cromossomos dos indivíduos, mantendo a diversidade genética.
- Seleção de Sobreviventes: Operadores de seleção de sobreviventes determinam quais indivíduos serão mantidos na próxima geração, com base em critérios como elitismo ou seleção por aptidão.

Os autores Silva e Ludermir (2021) empregam algoritmos genéticos para descobrir redes neurais compactas e de alto desempenho. Isso se deve à capacidade desses algoritmos de explorar o espaço de soluções de maneira eficaz, encontrando modelos com boas métricas em uma fração do conjunto de dados completo.

O trabalho de Lu et al. (2020b) apresentam uma proposta de algoritmo evolutivo para descobrir arquiteturas eficazes de redes neurais em múltiplos objetivos, superando tanto as arquiteturas projetadas manualmente quanto as automaticamente em conjuntos de dados de classificação de imagens de referência. Por outro lado, Chu et al. (2019) combinam Algoritmos Genéticos com Aprendizado por Reforço para mitigar a degradação dos modelos durante o processo de aprendizagem.

Todo estudo do estado da arte necessário para o desenvolvimento deste trabalho, que será apresentado no próximo capítulo, foi utilizado como revisão bibliográfica. Neste capítulo, são abordadas contribuições relevantes da literatura sobre os temas tratados neste trabalho. Primeiramente, é discutido o conceito de AutoML, uma abordagem que permite a construção de modelos de inteligência artificial sem intervenção humana. Destaca-se a importância da busca por arquiteturas de rede neural, especialmente por meio de técnicas como a NAS. Além disso, o capítulo explora como o AutoML pode ser aplicado em diversas etapas do Aprendizado de Máquina, incluindo a Otimização de Hiperparâmetros.

Dentro da Otimização de Hiperparâmetros, a abordagem multiobjetivo desempenha um papel crucial, permitindo a consideração de múltiplos critérios de desempenho. Destaca-se a importância de técnicas como a Otimização Multiobjetivo, que lida com problemas que envolvem a consideração de múltiplos objetivos, e a abordagem lexicográfica, que otimiza os objetivos de maneira sequencial, seguindo uma ordem de prioridade definida pelo usuário. O capítulo também explora o conceito de AG, uma técnica inspirada na evolução natural que reproduz o processo de seleção natural. Os AGs são frequentemente usados em conjunto com Redes Neurais para explorar de forma eficaz o espaço de soluções.

No próximo capítulo será abordado o desenvolvimento de um Algoritmo Genético que realiza a Otimização de Hiperparâmetros Multiobjetivo, utilizando da Fronteira de Pareto e da Abordagem Lexicográfica para encontrar a melhor solução de um problema dado de forma automática, minimizando a interferência humana.

3 DESENVOLVIMENTO

Neste capítulo será apresentada a metodologia adotada para o desenvolvimento do método lexicográfico genético multiobjetivo para redes profundas. Será destacado quais rotinas foram desenvolvidas e quais foram aproveitadas de pacotes e as razões para cada decisão tomada. Ao final, como forma de validar o desenvolvimento, escolheu-se um *dataset* popular para ter uma avaliação do desempenho e das potencialidades dessa abordagem desenvolvida.

O *dataset* escolhido para a realização do trabalho foi o MNIST que consiste em imagens de dígitos numéricos escritos manuscritos, desse conjunto de dados foram separadas 10000 imagens. O código de desenvolvimento encontra-se no repositório do github: <https://github.com/jpamatos/multiobjective>. Desenvolvido em Python, o código foi divido em 3 *scripts* principais: O *main*, que realiza o carregamento das imagens e contém as chamadas do *script* do algoritmo genético. O *script individual* que contém a classe indivíduo para a execução do algoritmo e o *script genetic_algorithm* que contém a classe do algoritmo genético.

As execuções foram realizadas no ambiente virtual *GoogleColab* e o *notebook(.ipynb)* resultante encontra-se no repositório.

3.1 O Script Main

Este *script* realiza o carregamento das imagens do conjunto de dados, utilizando os próprios dados fornecidos pelo pacote *Keras*. Uma das vantagens do uso do algoritmo genético é a capacidade de realizar experimentos em apenas uma parte do conjunto de dados. A rede neural pode ser treinada e otimizada em uma fração dos dados e, após a obtenção de um modelo otimizado, o treinamento pode ser continuado com o restante dos dados.

Neste experimento, as 10.000 imagens são divididas em 70% para treinamento e 30% para teste. A partir daí, o algoritmo genético é instanciado e executado. Isso resulta na obtenção da rede neural otimizada, a qual é salva em um arquivo do tipo *h5*. Esse arquivo contém as informações e os pesos do modelo, permitindo que o usuário o utilize conforme necessário.

3.2 A Classe Indivíduo

Para a execução do algoritmo genético, é necessário existir um objeto indivíduo para gerar a população e armazenar o modelo. Esse objeto é criado por esse *script*.

A classe *individual* contém a geração em que o indivíduo foi criado, as métricas de avaliação (os objetivos), o modelo de rede neural e o cromossomo que representa esse indivíduo. O cromossomo é uma lista, de tamanho 12, contendo valores 0 e 1, gerados de forma aleatória, que codifica as informações da arquitetura de rede neural em valores binários, onde:

- Os 2 primeiros genes codificam o número de camadas convolucionais, entre 1 e 4.
- Os próximos 2 genes codificam o número de neurônios nas camadas convolucionais, em potências de 2, entre 16 e 128.
- Os próximos 2 genes codificam o número de camadas densas no modelo, entre 1 e 4.
- Os próximos 6 genes, codificam o número de neurônios nas camadas densas, entre 1 e 64

Por exemplo, um indivíduo com o cromossomo: [0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1], resultaria em:

$$01_2 = 1_{10} + 1 = 2 \textit{ camadas convolucionais}$$

$$10_2 = 2_{10} = 2^{2+4} = 64 \textit{ neurônios em cada camada convolucional}$$

$$11_2 = 3_{10} + 1 = 4 \textit{ camadas densas}$$

$$011011_2 = 27_{10} + 1 = 28 \textit{ neurônios em cada camada densa}$$

O valor 1 é adicionado depois da codificação para evitar que sejam criadas redes neurais com 0 camadas ou camadas com 0 neurônios.

O modelo criado através do cromossomo é treinado com 20% dos dados de treinado separados para a validação, depois, avaliado no conjunto de teste. As métricas de avaliação são calculadas que formam o conjunto de objetivos a serem otimizados e foram escolhidas porque cada uma delas oferece uma perspectiva valiosa sobre o desempenho do modelo de aprendizado de máquina, são elas:

- *loss*: É uma função que quantifica a diferença entre as previsões do modelo e os valores reais do conjunto de dados de treinamento. O objetivo do treinamento da rede neural é minimizar essa função de perda, ajustando os pesos da rede através de algoritmos de otimização, como o gradiente descendente, para melhorar a precisão das previsões.

- acurácia: É uma métrica que mede a proporção de previsões corretas feitas pelo modelo em relação ao total de previsões. Em outras palavras, é a medida de quão precisas são as previsões do modelo.
- norma dos pesos: É a norma dos pesos que se refere à magnitude dos pesos atribuídos às conexões entre os neurônios em uma rede neural. A norma dos pesos é importante porque pode ajudar a regularizar o modelo.
- Latência: É a latência que se refere ao tempo necessário para que a rede processe uma entrada e gere uma saída. É uma medida importante em aplicações em tempo real, onde a velocidade de resposta é crítica.
- *f1_score*: É uma métrica que leva em consideração tanto a precisão (quantos dos itens identificados como pertencentes a uma classe pertencem realmente a ela) quanto o *recall* (quantos dos itens que pertencem a uma classe foram corretamente identificados pelo modelo). O F1 Score é a média harmônica dessas duas métricas e é útil quando há um desequilíbrio significativo nas classes de dados.

A classe indivíduo ainda contém o método *crossover*, que gera dois filhos a partir de dois indivíduos, fazendo o cruzamento dos genes e também a função de mutação, que inverte o valor de um bit do cromossomo com uma probabilidade determinada.

3.3 A Classe Algoritmo Genético

A classe criada pelo *script genetic_algorithm* realiza o processo de otimização. Primeiramente, o objeto precisa receber o conjunto de dados e os objetivos juntamente com a tolerância de cada um. Para executar o algoritmo, é necessário chamar o método solve e indicar o número de gerações e a taxa de mutação.

O algoritmo inicia a execução criando uma lista de indivíduos aleatórios de tamanho definido pelo usuário, chamando essa geração de geração 0. Cada objeto indivíduo que foi inicializado é executado pela chamada da função de avaliação e as métricas são armazenadas no objeto do indivíduo.

Então, para cada geração, até o número de gerações máximo, a execução se repete. O laço de repetição se inicia encontrando a Fronteira de Pareto da população. Como visto na Seção 2.1, uma solução x, será dominada por todas as soluções do conjunto, se para toda solução y.

$$\forall y | f(y) \leq f(x) \tag{3.1}$$

Como queremos que x seja uma solução não dominada, a proposição 3.1 é negada, como o problema é multiobjetivo com 2 objetivos, encontramos a proposição 3.2, onde

se não existir um y que a solução x tenha ao menos um objetivo maior que ela, x é uma solução não dominada.

$$\nexists y | f_1(x) > f_1(y) \ \wedge \ f_2(x) > f_2(y) \tag{3.2}$$

O conjunto de soluções Pareto Ótimo é formado por soluções onde essa proposição é válida.

Após encontrar o Conjunto Pareto Ótimo são realizadas as operações de cruzamento e mutação, onde os pais são escolhidos conforme o critério de preferência de pertencer ao Conjunto Pareto Ótimo daquela geração, se não houver pais suficientes no conjunto eles serão escolhidos de soluções dominadas.

A melhor solução da geração, é encontrada através da Abordagem Lexicográfica, como visto na Seção 2.1.2. Para as soluções no Conjunto Pareto, as métricas são avaliadas conforme o método Lexicográfico.

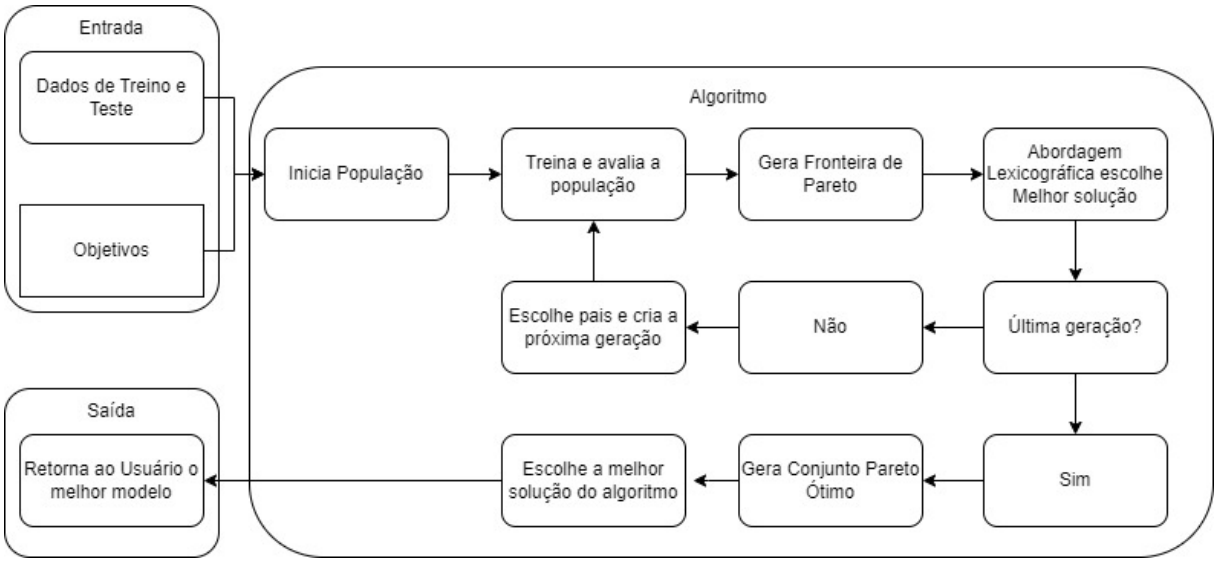
- Primeiro é analisado o primeiro objetivo, escolhendo a solução com a melhor métrica.
- O limiar, definido anteriormente, irá determinar a faixa de escolha do primeiro conjunto, ele define qual a porcentagem acima da melhor métrica será aceita para ir para o próximo passo. As soluções que estiverem dentro dessa formam o conjunto de melhores soluções para o primeiro objetivo.
- Se apenas a uma solução estiver dentro desse conjunto, ela é escolhida, como a melhor.
- Caso haja mais soluções, o mesmo é feito para o segundo objetivo e as soluções dentro desse subconjunto, criando um segundo subconjunto limitado pela melhor solução para o segundo objetivo e a porcentagem aceitável acima dela. Criando assim, o conjunto de melhor soluções entre os dois objetivos.
- Se apenas uma solução, a melhor no segundo objetivo, estiver nesse subconjunto, ela é escolhida.
- Se houver mais de uma solução nele, a solução que apresentar o melhor primeiro objetivo é escolhida.

A execução do algoritmo prossegue até alcançar o fim das gerações, quando o conjunto Pareto final é definido e a melhor solução, o modelo otimizado, é escolhido.

O algoritmo calcula o Conjunto Pareto Ótimo através do conjunto de soluções Parto formado ao longo da execução, todas essas soluções são analisadas da mesma forma que anteriormente e um Conjunto Pareto final é encontrado. Para escolher o melhor modelo é

utilizado o método Lexicográfico nesse conjunto, que escolherá o melhor modelo conforme os objetivos, a figura mostra um diagrama esquemático da execução do algoritmo.

Figura 3 – Diagrama de execução do algoritmo



Fonte: Elaborado pelo autor

Pela Figura 3, o algoritmo recebe os dados de treino e de teste, juntamente com os objetivos a serem otimizados e seus limiares em porcentagem decimal. A primeira população na geração 0 é inicializada com cromossomos aleatórios, nela, cada indivíduo é treinado e avaliado. As métricas da avaliação são utilizadas para gerar a conjunto Pareto dessa população. Esse conjunto Pareto é armazenado em uma lista referente ao conjunto Pareto de todas as populações. Desse Pareto são escolhidos os pais da próxima geração conforme a disponibilidade, se não houver o suficiente, a escolha é feita no resto da população. Ao mesmo tempo, a melhor solução do algoritmo é escolhida e atualizada conforme a abordagem Lexicográfica. O algoritmo continua a execução até o fim das gerações, onde a lista que armazenou todos os conjunto Pareto ao longo da execução é analisada e o Conjunto Pareto Ótimo é calculado. Desse conjunto a melhor solução é escolhida mais uma vez através da abordagem Lexicográfica. Esse modelo, é o melhor modelo resultante dos objetivos escolhidos e os limiares, ele então é retornado ao usuário em forma de um arquivo *.keras*, para poder ser usado conforme a necessidade.

No próximo capítulo será abordado os testes realizados nesse algoritmo, onde foram criadas três situações hipotéticas e observadas como os objetivos escolhidos afetariam a situação e se o resultado objetivo está conforme o esperado para o problema.

4 RESULTADOS

O Algoritmo Genético foi executado com o *dataset* MNIST em configurações diferentes, inspirado em 3 situações independentes. A tolerância que será enviada ao algoritmo para executar o método Lexicográfico foi escolhida de forma arbitrária. Na primeira configuração foram escolhidos os objetivos *loss* e norma dos pesos, a segunda configuração determinou os objetivos como acurácia a 5% de limiar e o tempo de inferência (latência) a 30%, já a terceira, finalizou os experimentos com os objetivos *loss* a 20% de tolerância e latência com os mesmos 20%.

Para cada experimento será exibido um gráfico contendo todas as soluções Pareto da execução do algoritmo e então será formado um Conjunto Pareto Ótimo da execução. As soluções em azul são soluções dominadas por outras soluções. As soluções em vermelho são soluções Não dominadas e, portanto, formam o Conjunto Pareto Ótimo, a solução em verde é a solução escolhida para ser a melhor do algoritmo, através do método Lexicográfico, levando em consideração os objetivos e os limiares.

4.1 Experimento 1: Loss x Norma dos Pesos

Sendo inspirados por um problema onde a perda durante o treino é a métrica mais importante a se otimizar, ou seja, requeremos um modelo muito preciso e útil, juntamente com a norma dos pesos como segundo objetivo, esse modelo também seria executado em um dispositivo com memória limitada, onde uma rede menor é crucial.

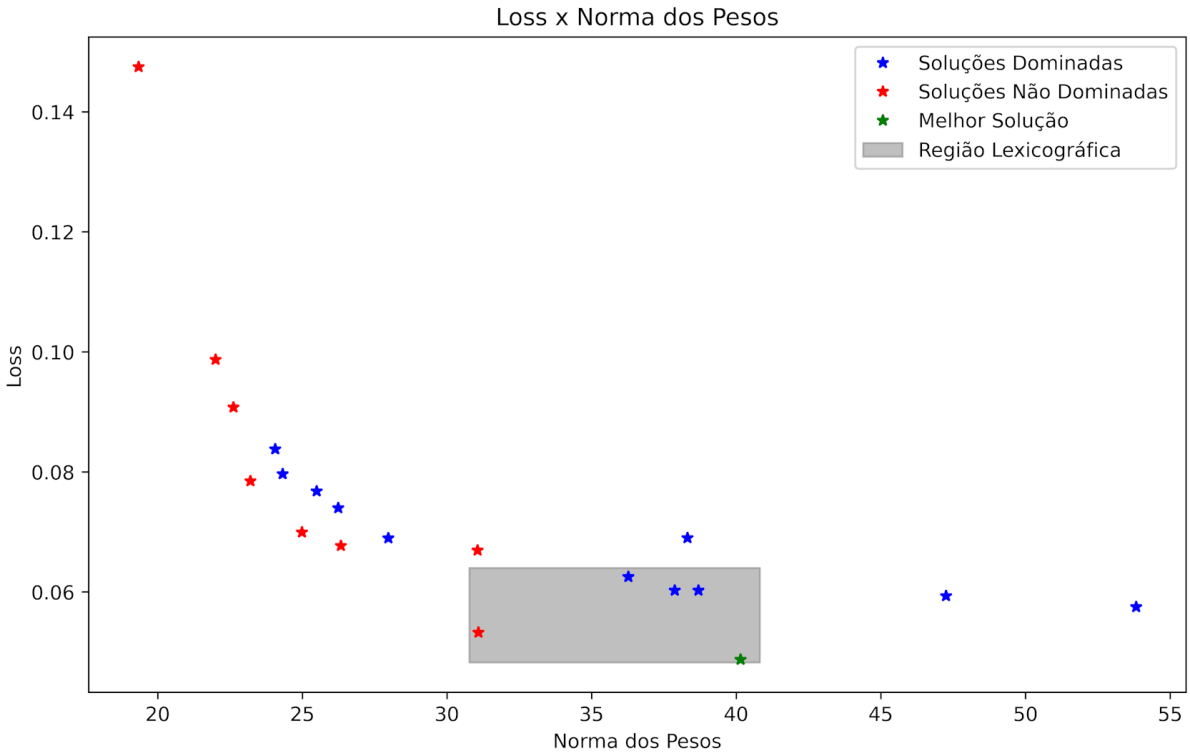
Esse teste foi executado com as métricas *loss*, com o limiar de 30% e norma dos pesos com limiar de 30%, o resultado desse teste está representado na Figura 4.

Onde os pontos azuis são as soluções dominadas, os pontos vermelhos as soluções não dominadas, formando o Conjunto Pareto Ótimo e o ponto verde, a região cinza é a região produzida pela Abordagem Lexicográfica e a solução em verde é a melhor solução escolhida por ela.

É possível observar que a solução escolhida não está na região inferior do Pareto, onde a métrica *loss* apresentou um bom desempenho e métrica norma dos pesos apresentou um desempenho aceitável. A Região Lexicográfica resultante contem mais de uma solução, então, o algoritmo optou pela que apresentou o menor primeiro objetivo (*loss*).

A tabela 2 mostra o resultado do algoritmo genético e o espaço de soluções, observamos as métricas do primeiro objetivo e do segundo objetivo das soluções não dominadas e dominadas, no fim da tabela encontra-se a solução escolhida e seus respectivos genomas. O modelo resultado possui o genoma que caracteriza em 4 camadas convolucionais

Figura 4 – Loss(0.3) x Norma dos Pesos(0.3)



Fonte: Elaborado pelo autor

Tabela 2 – Resultados Loss x Norma dos Pesos

Genoma	Primeiro Objetivo	Segundo Objetivo	Solução
1 0 0 0 0 0 1 0 0 1 0 1	0.078501239	23.20551333	não dominada
1 0 0 1 0 0 1 0 0 1 0 1	0.067721166	26.33114319	não dominada
1 0 0 0 0 0 1 0 0 0 0 0	0.098732911	22.00009184	não dominada
1 0 0 0 0 0 0 0 1 0 0 1	0.147518501	19.33358182	não dominada
1 0 0 0 0 0 1 0 0 0 0 1	0.09078835	22.61336391	não dominada
1 0 0 0 0 0 1 1 0 0 0 1	0.069972679	24.98496541	não dominada
1 0 1 0 0 0 1 0 0 1 0 0	0.066944741	31.05537451	não dominada
1 0 1 0 0 0 1 0 0 1 0 1	0.05327956	31.08643096	não dominada
1 1 1 0 0 0 1 1 0 0 0 1	0.04874007	40.14988914	solução escolhida

Fonte: Produzido pelo autor

com 64 neurônios e 1 camada densa com 50 neurônios.

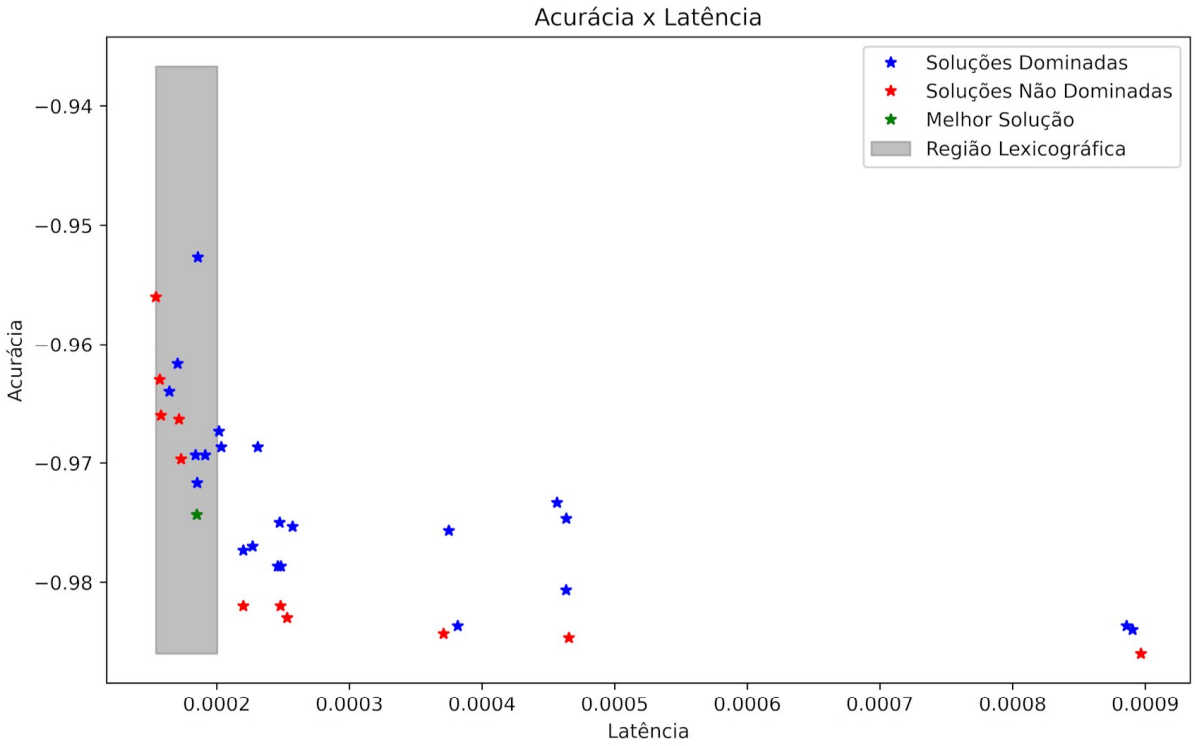
4.2 Experimento 2: Acurácia x Latência

Nessa situação, pensamos em um modelo que necessite de alta acurácia e também um tempo de inferência baixo, por exemplo, em um aplicativo que identifique que o usuário

esteja fazendo exercícios físicos para ativar outras funcionalidades do dispositivo, precisa apresentar uma alta acurácia para não ativá-las desnecessariamente, resultando em gasto de bateria. O tempo de inferência baixo representaria uma situação em que esse modelo seja executado em tempo real.

A Figura 5, contém esse teste, realizado-o com a acurácia (acurácia) a 5% de limiar e o tempo de inferência (latência) a 30%.

Figura 5 – Acurácia(0.05) x Latência(0.3)



Fonte: Elaborado pelo autor

Nesse caso, com apenas 5% de escolha para a acurácia, é possível observar muitas soluções na disputa da escolha, o fator limitante nesse caso foi a escolha do segundo objetivo, sendo possível observar que mais uma vez mais uma solução participa do conjunto delimitado pela Região Lexicográfica, o critério de desempate, mais uma vez, será o menor primeiro objetivo.

O genoma do melhor resultado, apresentado na tabela 3, resulta em um modelo com 3 camadas convolucionais com 16 neurônios e 2 camadas densas com 64 neurônios.

Um modelo com baixo tempo de inferência, também, resulta em um modelo menor, e mais importante ainda, um modelo em que não há muitos números de neurônios, principalmente em camas convolucionais que tendem a acrescentar o tempo de execução do

Tabela 3 – Resultados Acurácia x Latência

Genoma	Primeiro Objetivo	Segundo Objetivo	Solução
0 0 0 0 0 0 0 1 1 0 1 1	0.95599997	0.000154148	não dominada
1 1 1 0 0 1 0 1 1 1 1 1	0.984666646	0.000465475	não dominada
1 1 0 1 1 1 1 1 1 0 0 0	0.98299998	0.000253145	não dominada
0 0 0 0 1 1 1 1 1 0 0 0	0.96633333	0.000171607	não dominada
1 0 0 1 0 1 1 1 1 1 1 1	0.981999993	0.000220057	não dominada
0 1 0 0 0 1 0 1 1 1 1 1	0.963	0.000157016	não dominada
1 1 1 1 0 1 1 1 1 1 1 1	0.986000001	0.000896753	não dominada
1 0 1 0 0 1 1 1 1 1 1 1	0.984333336	0.000370981	não dominada
0 0 0 0 0 0 0 1 1 0 0 0	0.966000021	0.000157954	não dominada
1 0 0 1 0 1 1 1 1 0 0 0	0.981999993	0.000248101	não dominada
0 0 0 0 1 1 1 1 1 1 0 0	0.96966666	0.00017309	não dominada
1 0 0 0 0 1 1 1 1 1 1 1	0.974333346	0.000185048	solução escolhida

Fonte: Produzido pelo autor

modelo. Otimizando esse tempo de execução enquanto da prioridade na acurácia, evidencia um modelo interessante para a situação ponderada.

4.3 Experimento 3: Loss x Latência

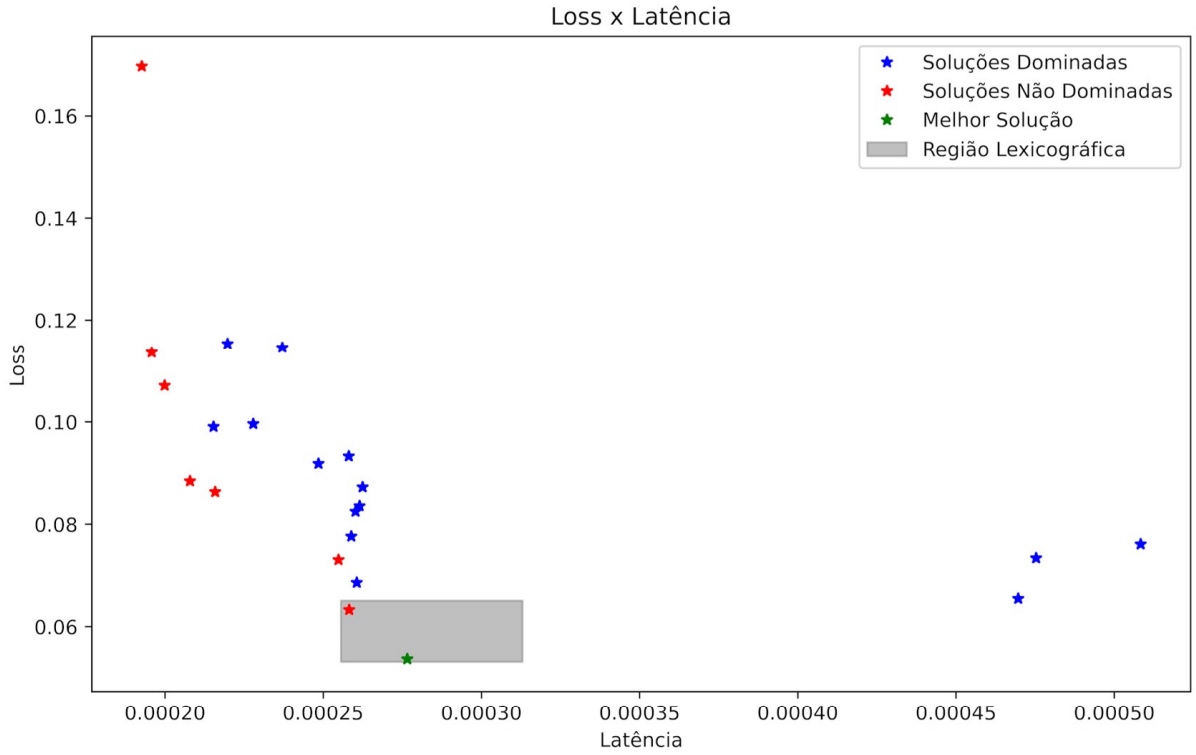
Nesse teste podemos pensar em uma situação em que o modelo será usado em um projeto na área da saúde, onde uma detecção de alguma complicação de saúde ocorra de forma precisa e em tempo real, como uma crise iminente onde o indivíduo necessite estar preparado ou em algum local seguro.

Executamos o teste com os objetivos *loss* a 20% de tolerância e latência com os mesmos 20%, para conseguirmos um modelo que aprenda bem o conjunto de dados e seja executado de forma rápida. A Figura 6 mostra o resultado desse experimento.

Nesse caso, o limiar da latência não permitiu que a solução com a menor *loss* fosse escolhida, e permitiu que uma solução com um *tradeoff* melhor entre os objetivos fosse escolhida.

O melhor modelo retornado pelo algoritmo, exibido na tabela 4, com genoma que caracteriza 4 camadas convolucionais com 32 neurônios e 3 camadas densas com 64 neurônios. Resulta em um baixo tempo de execução juntamente com a baixa *loss* que caracteriza um bom modelo para a nossa situação. É evidente o resultado de um tempo de execução maior que o resultante do teste 2, o que é esperando devido que a maior contribuição para aumentar o tempo de execução vem das camadas convolucionais.

Figura 6 – Loss(0.2) x Latência(0.2)



Fonte: Elaborado pelo autor

Tabela 4 – Resultados Loss x Latência

Genoma	Primeiro Objetivo	Segundo Objetivo	Solução
1 0 0 1 1 1 1 0 1 0 1 1	0.073080339	0.000254796	não dominada
1 0 0 1 1 0 1 1 1 1 0 1	0.063320406	0.000258202	não dominada
1 0 0 0 1 0 1 0 1 1 1 1	0.088423237	0.00020791	não dominada
0 1 0 0 0 0 1 1 1 1 1 1	0.113622226	0.000195807	não dominada
1 0 0 0 1 0 1 0 1 0 0 0	0.086333424	0.000215852	não dominada
0 1 0 0 0 0 1 1 1 1 0 1	0.107112937	0.000199899	não dominada
0 1 0 0 0 0 0 0 1 0 0 1	0.169743046	0.000192629	não dominada
1 1 0 1 1 0 1 1 1 1 1 1	0.053690776	0.000276569	solução escolhida

Fonte: Produzido pelo autor

Com os resultados dos três experimentos, fica evidente que o algoritmo foi bem-sucedido em identificar a melhor configuração para cada situação, levando em conta os objetivos estabelecidos e respeitando os limiares definidos. Notamos também que simplesmente ajustar o número de camadas e neurônios pode resultar em modelos com métricas bastante divergentes, o que, no nosso caso, está alinhado com a natureza de cada

problema que foi abordado.

5 CONCLUSÃO

Ao concluir este estudo sobre o treinamento de redes profundas utilizando abordagens multiobjetivo lexicográficas, podemos destacar diversos pontos significativos. Inicialmente, desenvolvemos e implementamos um algoritmo genético multiobjetivo que permite o treinamento de redes profundas para otimização de diferentes objetivos simultaneamente. A utilização dessa abordagem multiobjetivo permite uma maior flexibilidade e adaptabilidade do modelo às necessidades específicas de diferentes problemas.

Além disso, ao implementar uma abordagem lexicográfica para direcionar a busca por soluções preferenciais, simplificamos o processo de tomada de decisão para o usuário, permitindo uma compreensão mais clara da importância dos diferentes objetivos envolvidos no treinamento da rede neural, juntamente com os limiares de cada objetivo para ser possível adaptar o problema para a necessidade específica do usuário. Este estudo oferece uma contribuição significativa para o campo do treinamento de redes profundas, apresentando uma abordagem eficaz para lidar com múltiplos objetivos de forma simultânea.

Os resultados da otimização multiobjetivo nos problemas abordados revelaram-se promissores, e os testes realizados confirmaram o comportamento esperado. O algoritmo foi capaz de identificar soluções pertencentes ao Conjunto Pareto Ótimo, que são não dominadas, demonstrando sua capacidade de encontrar um equilíbrio entre os diferentes objetivos considerados. A utilização da Abordagem Lexicográfica para selecionar a melhor solução foi satisfatória, restringindo efetivamente a região de busca e garantindo que não fossem selecionadas soluções onde uma métrica fosse significativamente pior em relação à outra. É importante ressaltar que a seleção dos objetivos desempenha um papel crucial nesse processo, influenciando diretamente nos resultados obtidos.

No primeiro cenário, em que otimizamos a *loss* e a norma dos pesos, obtivemos um modelo com um número reduzido de neurônios e uma baixa *loss*. Esse modelo é ideal para situações em que se requer alta precisão com restrições de memória, sendo adequado para execução em dispositivos com recursos limitados. No segundo caso, ao focarmos na otimização da acurácia e da latência, conseguimos desenvolver um modelo com um tempo de inferência extremamente baixo, tornando-o ideal para aplicações em tempo real. Este modelo conseguiu manter uma alta acurácia, mesmo em comparação com outros modelos mais rápidos. Na terceira situação, em que buscamos otimizar a *loss* e a latência, obtivemos um modelo que combina rapidez e precisão, semelhante ao segundo cenário. Esses resultados destacam a versatilidade das abordagens multiobjetivo para a obtenção de modelos que atendem a diferentes requisitos de desempenho e aplicação.

Como trabalhos futuros, existe a possibilidade de adicionar mais parâmetros para

de otimizar na rede neural, como, por exemplo, função de ativação, otimizador, *scaler*. Também é possível utilizar algoritmo para criar outros tipos de redes neurais e resolver outros problemas, fora da área de classificação de imagens. Ou até criar um algoritmo mais genérico, suportando todos os tipos de camadas de redes neurais.

Referências

BROOKHOUSE, J.; FREITAS, A. *Fair Feature Selection: A Comparison of Multi-Objective Genetic Algorithms*. 2023. Citado na página 20.

CHAGAS, E. T. D. O. Deep learning e suas aplicações na atualidade. *Revista Científica Multidisciplinar Núcleo do Conhecimento*, Núcleo do Conhecimento, v. 4, n. 5, p. 5–26, 2019. ISSN 2448-0959. Disponível em: <https://www.nucleodoconhecimento.com.br/administracao/deep-learning>. Citado na página 12.

CHEN, Z. et al. *Multi-objective Neural Architecture Search via Non-stationary Policy Gradient*. 2020. Citado na página 18.

CHU, X. et al. *Multi-Objective Reinforced Evolution in Mobile Neural Architecture Search*. 2019. Citado na página 22.

CONG, S.; ZHOU, Y. A review of convolutional neural network architectures and their optimizations. *Artificial Intelligence Review*, v. 56, n. 3, p. 1905–1969, Mar 2023. ISSN 1573-7462. Disponível em: <https://doi.org/10.1007/s10462-022-10213-5>. Citado na página 18.

ELSKEN, T.; METZEN, J. H.; HUTTER, F. *Efficient Multi-objective Neural Architecture Search via Lamarckian Evolution*. 2019. Citado na página 18.

FALANTI, A. et al. Popnasv2: An efficient multi-objective neural architecture search technique. In: *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022. Disponível em: <http://dx.doi.org/10.1109/IJCNN55064.2022.9892073>. Citado na página 18.

FREITAS, A. A. A critical review of multi-objective optimization in data mining: a position paper. *SIGKDD Explor. Newsl.*, Association for Computing Machinery, New York, NY, USA, v. 6, n. 2, p. 77–86, dec 2004. ISSN 1931-0145. Disponível em: <https://doi.org/10.1145/1046456.1046467>. Citado 3 vezes nas páginas 16, 18 e 19.

GUERRERO-VIU, J. et al. *Bag of Baselines for Multi-objective Joint Neural Architecture Search and Hyperparameter Optimization*. 2021. Citado na página 16.

HASHIMOTO, K. *Técnicas de otimização combinatória multiobjetivo aplicadas na estimação do desempenho elétrico de redes de distribuição*. Tese (Doutorado) — Escola Politécnica, Universidade de São Paulo, São Paulo, 2004. Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3139/tde-19112004-165342/>. Citado na página 17.

HE, X.; ZHAO, K.; CHU, X. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, Elsevier BV, v. 212, p. 106622, jan. 2021. ISSN 0950-7051. Disponível em: <http://dx.doi.org/10.1016/j.knosys.2020.106622>. Citado 3 vezes nas páginas 12, 15 e 16.

HSU, C.-H. et al. *MONAS: Multi-Objective Neural Architecture Search using Reinforcement Learning*. 2018. Citado na página 15.

HU, S. et al. *Multi-objective Neural Architecture Search with Almost No Training*. 2020. Citado na página 18.

HU, S. et al. *Accelerating Multi-Objective Neural Architecture Search by Random-Weight Evaluation*. 2021. Citado na página 18.

JENTZEN, A.; KUCKUCK, B.; WURSTEMBERGER, P. von. *Mathematical Introduction to Deep Learning: Methods, Implementations, and Theory*. 2023. Citado na página 12.

LINCZUK, V. C. C.; BASTOS, L. E. G. Otimização multiobjetivo orientada ao desempenho térmico para o projeto de edificações de baixo consumo de energia na região sul do brasil. *Ambiente Construído*, v. 20, n. 4, p. 509–529, out. 2020. Disponível em: <https://seer.ufrgs.br/index.php/ambienteconstruido/article/view/95403>. Citado na página 17.

LU, Z. et al. *Neural Architecture Search as Multiobjective Optimization Benchmarks: Problem Formulation and Performance Assessment*. 2023. Citado na página 18.

LU, Z. et al. *NSGANetV2: Evolutionary Multi-Objective Surrogate-Assisted Neural Architecture Search*. 2020. Citado na página 18.

LU, Z. et al. *Multi-Objective Evolutionary Design of Deep Convolutional Neural Networks for Image Classification*. 2020. Citado na página 22.

MAHESH, K.; NALLAGOWNDEN, P.; ELAMVAZUTHI, I. Advanced pareto front non-dominated sorting multi-objective particle swarm optimization for optimal placement and sizing of distributed generation. *Energies*, v. 9, n. 12, 2016. ISSN 1996-1073. Disponível em: <https://www.mdpi.com/1996-1073/9/12/982>. Citado na página 19.

PACHECO, M. A. C. *ALGORITMOS GENÉTICOS: PRINCÍPIOS E APLICAÇÕES*. 2004. Notas de Aula. Disponível em: <https://www.inf.ufsc.br/~mauro.roisenberg/ine5377/Cursos-ICA/CE-intro_apost.pdf>. Citado na página 21.

PFISTERER, F. et al. *Multi-Objective Automatic Machine Learning with AutoxgboostMC*. 2021. Citado na página 18.

QIAN, F. et al. Overview of multiobjective particle swarm optimization algorithm. *Chinese Journal of Engineering*, v. 43, n. 6, p. 745–753, 2021. ISSN 2095-9389. Disponível em: <http://cje.ustb.edu.cn/en/article/doi/10.13374/j.issn2095-9389.2020.10.31.001>. Citado na página 16.

RASTEGAR, N.; KHORRAM, E. Relaxation of constraints in lexicographic multiobjective programming problems. *Optimization*, Taylor Francis, v. 64, n. 10, p. 2111–2129, 2015. Disponível em: <https://doi.org/10.1080/02331934.2014.929785>. Citado na página 20.

SILVA, A. da; LUDERMIR, T. *Otimizacao de Redes Neurais atraves de Algoritmos Geneticos Celulares*. 2021. Citado 2 vezes nas páginas 21 e 22.

TRUSS, M.; SCHMITT, M. *Human-Centered AI Product Prototyping with No-Code AutoML: Conceptual Framework, Potentials and Limitations*. 2024. Citado na página 12.

WANG, C. et al. *Multi-Objective Neural Architecture Search Based on Diverse Structures and Adaptive Recommendation*. 2020. Citado na página 18.

WISTUBA, M.; RAWAT, A.; PEDAPATI, T. *A Survey on Neural Architecture Search*. 2019. Citado na página 15.