UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS
COLEGIADO DO CURSO DE ENGENHARIA DE CONTROLE
E AUTOMAÇÃO - CECAU

ANTONIO DE BARROS NADDEO MEIRELLES FERREIRA

OTIMIZAÇÃO MULTIOBJETIVO BASEADA EM MODELOS SUBSTITUTOS PARA COMPRESSÃO DE REDES NEURAIS ARTIFICIAIS

MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Ouro Preto, 2022

ANTONIO DE BARROS NADDEO MEIRELLES FERREIRA

OTIMIZAÇÃO MULTIOBJETIVO BASEADA EM MODELOS SUBSTITUTOS PARA COMPRESSÃO DE REDES NEURAIS ARTIFICIAIS

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Prof. Rodrigo César Pedrosa Silva, Ph.D.

Coorientador: Prof. Agnaldo José da Rocha Reis, Dr.

Ouro Preto
Escola de Minas – UFOP
2022

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

F383o Ferreira, Antonio de Barros Naddeo Meirelles.
Otimização multiobjetivo baseada em modelos substitutos para compressão de redes neurais artificiais. [manuscrito] / Antonio de Barros Naddeo Meirelles Ferreira. - 2022.
43 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Rodrigo César Pedrosa Silva.
Coorientador: Prof. Dr. Agnaldo José da Rocha Reis.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Escola de Minas. Graduação em Engenharia de Controle e Automação .

1. Redes Neurais (Computação). 2. Redes neurais-Compressão. 3. Otimização. I. Reis, Agnaldo José da Rocha. II. Silva, Rodrigo César Pedrosa. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 621.31

Bibliotecário(a) Responsável: Angela Maria Raimundo - SIAPE: 1.644.803

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO

ATA DA SESSÃO DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

No sétimo dia do mês de janeiro de dois mil e vinte dois, realizou-se às 08 horas, de forma não presencial, por meio do aplicativo Google Meet (meet.google.com/ndd-qmhj-yay), a sessão de defesa de Trabalho de Conclusão de Curso do candidato ao grau de Engenheiro de Controle e Automação, Antonio de Barros Naddeo Meirelles Ferreira, intitulada "OTIMIZAÇÃO MULTIOBJETIVO BASEADA EM MODELOS SUBSTITUTOS PARA COMPRESSÃO DE REDES NEURAIS ARTIFICIAIS". A Banca Examinadora foi constituída por Rodrigo Cesar Pedrosa Silva (Orientador, UFOP/ICEB/DECOM), Agnaldo José da Rocha Reis (Coorientador, UFOP/EM/DECAT), Tamires Martins Rezende (FITec) e Gabriel Lima de Souza (Doutorando PPGCC/UFOP). O Prof. Rodrigo abriu a sessão agradecendo a participação dos examinadores supracitados e passou a palavra ao candidato, que fez a exposição do seu trabalho. Em seguida, foi realizada a arguição pelos examinadores, com a respectiva defesa do candidato. Finalizada a arguição, a Banca Examinadora, sem a presença do candidato, deliberou pela sua **Aprovação**. Nada mais havendo para constar, lavrou-se a presente ata que será assinada eletronicamente pelos orientadores do trabalho via SEI!UFOP em nome de todos os membros da banca.

Documento assinado eletronicamente por **Agnaldo Jose da Rocha Reis**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 07/01/2022, às 09:18, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rodrigo Cesar Pedrosa Silva**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 07/01/2022, às 10:02, 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 **0265477** e o código CRC **9FDA9ED3**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar - Ouro Preto/MG, CEP 35400-000
Telefone: 3135591533 - www.ufop.br

Este trabalho é dedicado aos adultos que,
mantém o coração curioso de uma criança
e vivem como jovens dinâmicos.

AGRADECIMENTOS

Há uma frase atribuída a Newton que diz, “Se eu vi mais longe, foi por estar sobre ombros de gigantes.” E se hoje você lê este trabalho é devido também ao caminho que estes gigantes trilharam antes de mim.

Entretanto, esta frase ainda não contempla os gigantes que estiveram ao meu lado, e a estes eu dedico esse trabalho. Aos meus irmãos e irmãs da equipe 42, aos amigos de faculdade, agradeço pelo apoio durante as dificuldades na universidade, e por compartilharem durante os últimos anos comemorações, preocupações e despreocupações. Vocês fizeram meus dias mais felizes, muito obrigado senhores.

À equipe 12 bis e ao TerraLAB agradeço pelos grandes amigos que fiz, as ótimas lembranças e pelos aprendizados para vida.

Agradeço infinitamente a minha família, meu pai e minha mãe que me apoiam em minhas decisões, são minha base na vida, minha fonte de alegria e um farol quando me sinto perdido.

Este trabalho foi possível principalmente pelo meu orientador, professor Rodrigo, o qual me ensinou mais do que o currículo de suas matérias compreende. Me guiou excepcionalmente durante o desenvolvimento deste projeto e hoje tenho a felicidade em dizer que é um dos maiores amigos que a universidade me proporcionou. Agradeço também a todos os amigos que estiveram comigo de longe durante a pandemia, foi um período complicado e difícil, mas graças a vocês foi possível suportar as dificuldades deste tempo. E finalmente agradeço a Deus por me dar a possibilidade de conhecer pessoas tão maravilhosas, e me permitir diversas experiências e oportunidades que formaram a pessoa que sou.

“Matéria é a parte acidental.” (Oliver Lodge)

RESUMO

Nos últimos anos surgiram novos trabalhos visando a redução do custo computacional das redes neurais artificiais, mantendo a eficácia dos modelos. A redução deste custo pode permitir a aplicação de redes neurais artificiais em sistemas com restrições de hardware ou a redução de latência entre serviços da nuvem.

Entre os métodos criados, existem aqueles que buscam reduzir o custo de uma rede existente, como a poda de parâmetros ou quantização de pesos. Estes métodos transformam a arquitetura de uma rede, de forma a torná-la mais leve. Entretanto, estes métodos requerem configurações e hiper-parâmetros próprios para serem aplicados, configurações estas que normalmente são pouco discutidas na literatura. Aproveitando desta consideração, propõe-se neste trabalho uma metodologia para a otimização da escolha dos parâmetros envolvidos na compressão de uma rede neural artificial através de modelos substitutos. Esta técnica de otimização já se mostrou promissora em outros problemas semelhantes e pode ser uma alternativa interessante para a analise do problema de compressão.

Palavras-chaves: Redes Neurais Artificiais, Modelos Substitutos, Compressão de redes neurais, Otimização.

ABSTRACT

In recent years, new studies have emerged with the aim of reducing the computational cost of artificial neural networks, while maintaining the efficiency of the models. Reducing this cost allow the application of artificial neural networks in systems with hardware restrictions or the reduction of latency between cloud services.

Among the methods created, there are those that seek to reduce the cost of an existing network, such as pruning parameters or quantizing weights. These methods transform the architecture of a network in order to make it lighter. However, these methods require their own configurations and hyper-parameters, configurations that are usually little discussed in the literature. Taking advantage of this consideration, this work proposes an algorithm for optimizing the choice of parameters involved in the compression of an artificial neural network through surrogate models. This optimization technique has already shown good results in similar problems and can be an interesting alternative for analyzing the compression problem.

Key-words: Artificial Neural Networks, Surrogate Models, Neural Networks Compression, Optimization.

LISTA DE ILUSTRAÇÕES

Figura 1 – Representação de um neurônio artificial 18
Figura 2 – Representação de um rede neural artificial. 19
Figura 3 – Representação de uma camada convolucional. 20
Figura 4 – Representação do funcionamento de camada de *pooling*. 20
Figura 5 – Fluxograma do treinamento em lote de uma rede neural artificial. 21
Figura 6 – Esparsidade de uma rede ao longo do treinamento com aplicação do cronograma por esparsidade constante e decaimento polinomial. 24
Figura 7 – Representação da relação de dominância de Pareto. 26
Figura 8 – Representação da fronteira de Pareto. 26
Figura 9 – Hiper-volume de um problema biobjetivo. 27
Figura 10 – Diagrama de caixa dos hiper-volumes calculados para solução através de modelos substitutos do problema ZDT2. 34
Figura 11 – Diagrama de caixa dos hiper-volumes calculados para solução através de modelos substitutos do problema ZDT3. 35
Figura 12 – Diagrama de caixa dos hiper-volumes calculados para solução através de modelos substitutos do problema ZDT6. 35
Figura 13 – Fronteira de Pareto para problema de compressão da Rede ResNet50 através de *NSGA-II* e otimização por modelos substitutos. 37
Figura 14 – Fronteira de Pareto para o problema de compressão da Rede VGG16 através de *NSGA-II* e otimização por modelos substitutos. 38
Figura 15 – Soluções de compressão ResNet50 por método aplicado. 38
Figura 16 – Soluções de compressão VGG16 por método aplicado. 39
Figura 17 – Soluções de compressão ResNet50 por cronograma de poda. 40
Figura 18 – Soluções de compressão VGG16 por cronograma de poda. 40
Figura 19 – Soluções de compressão ResNet50 por frequência de poda. 41
Figura 20 – Soluções de compressão VGG16 por frequência de poda. 41

LISTA DE TABELAS

Tabela 1 – Variáveis de projeto. 31
Tabela 2 – Parâmetros utilizados na otimização por *NSGA-II*. 33
Tabela 3 – Parâmetros utilizados para algoritmo de otimização por modelos substitutos. *Parâmetros referentes a otimização do modelo substituto por *NSGA-II*. . . 33
Tabela 4 – Parâmetros utilizados na otimização dos parâmetros de compressão por modelos substitutos 33

LISTA DE ABREVIATURAS E SIGLAS

RNA	Redes Neurais Artificiais
NSGA-II	Non-dominated Sorting Genetic Algorithm II
HV	Hiper-Volume
RD	Seleção aleatória de *infill points*
SD	Seleção de *infill points* por distância no espaço de busca
OD	Seleção de *infill points* por densidade no espaço de objetivos
MSE	Erro quadrático médio

SUMÁRIO

1 INTRODUÇÃO 14
1.1 Objetivos gerais e específicos 16
1.2 Organização do texto 16

2 REDES NEURAIS ARTIFICIAIS E MÉTODOS DE COMPRESSÃO . 18
2.1 Redes Neurais Artificiais 18
2.1.1 Redes Convolucionais 19
2.1.2 Treinamento de uma rede neural artificial 20
2.2 Métodos de compressão de Redes Neurais 21
2.2.1 Poda 22
2.2.1.1 Cronograma por esparsidade constante 23
2.2.1.2 Cronograma por Decaimento Polinomial 23
2.2.2 Quantização 23
2.2.3 O custo da compressão 23

3 OTIMIZAÇÃO 25
3.1 Otimização Multiobjetivo 25
3.1.1 Hiper-Volume 25
3.2 Problema de otimização modelos substitutos 27
3.2.1 Algoritmo de otimização por modelos substitutos 28
3.2.2 Métodos de seleção de infill points 28
3.2.2.1 Seleção Aleatória (RD) 28
3.2.2.2 Distância no espaço de busca (SD) 29
3.2.2.3 Densidade no espaço de objetivos (OD) 29

4 METODOLOGIA 30
4.1 O problema de compressão de Redes Neurais Artificiais 30
4.2 Implementação do Algoritmo Proposto 31
4.2.1 Implementação do problema de compressão 31
4.2.2 Implementação do otimizador por modelos substitutos 32
4.3 Experimentos Computacionais 32
4.3.1 Experimento de Benchmark 32
4.3.2 Experimento de Compressão 33

5 RESULTADOS 34

5.1 Experimentos de Benchmark 34
5.2 Experimentos de Compressão de Redes Neurais Artificiais 37
5.2.1 Fronteira de Pareto *37*
5.2.2 Soluções por método *38*
5.2.3 Cronogramas de Poda *39*
5.2.4 Frequência de Poda *40*

6 CONCLUSÃO 42
6.1 Trabalhos Futuros 42

REFERÊNCIAS 43

1 INTRODUÇÃO

Aprendizado de máquina é a área que estuda e desenvolve métodos, algoritmos e modelos estatísticos que permitem à sistemas computacionais realizar tarefas sem instruções explicitas (MURPHY, 2013).

Apesar do aprendizado de máquina englobar diferentes ramos, cada um com suas peculiaridades, as redes neurais artificiais (RNA) e, em especial, a técnica de *deep learning* (GOODFELLOW; BENGIO; COURVILLE, 2016) são provavelmente as de maior destaque atualmente.

As RNA são formadas pelos chamados neurônios, ou nós. Cada nó é conectado a outros dos quais recebe ou para os quais envia informações. Estas conexões definem os pesos pelos quais o sinal de um neurônio deve ser multiplicado. Assim, o processo de treinamento de uma rede neural consiste em encontrar o melhor valor para estes pesos. As redes mais complexas, criadas por técnicas de *deep learning*, possuem diversas camadas, cada uma formada por um número variável de nós, indo de milhares a milhões de nós interligados (GOODFELLOW; BENGIO; COURVILLE, 2016). Assim sendo, redes neurais profundas possuem um grande número de parâmetros, como pesos e funções de ativação, necessários para inferência.

Estas redes neurais, em conjunto com outras técnicas, tem aplicações em diversas áreas. Elas podem, por exemplo, ser empregadas na navegação de robôs e carros autônomos, manipulação de objetos, detecção de erros em linhas de produção automatizadas e, recentemente, até no auxílio do diagnóstico da COVID-19 (LUZ et al., 2020).

Com os últimos avanços nestas técnicas, se observa um crescimento exponencial no número dos parâmetros destas redes (GOODFELLOW; BENGIO; COURVILLE, 2016). O aumento da precisão das RNAs é inegável, no entanto esta melhora demanda cada vez mais processamento e memória. Isso torna difícil a sua utilização em sistemas onde há restrições de hardware como microcontroladores e até *smartphones*.

Para a solução deste e outros problemas que envolvem o tamanho das RNAs, surgiram, nos últimos anos, esforços para a criação de redes neurais eficazes e compactas, através de novas arquiteturas ou por métodos de compressão de modelos.

Os métodos de compressão mais conhecidos, são a quantização e a poda. A quantização (WU et al., 2020) é o processo de redução da precisão dos valores de parâmetros, pesos ou ativações da rede neural. O método reduz tanto a memória necessária quanto o tempo de inferência. Enquanto a poda (ZHU; GUPTA, 2017) consiste no corte de conexões com pouca relevância para a inferência da rede.

Os trabalhos de He et al. (2018), Liu et al. (2017) propõem métodos para escolha e poda

de camadas convolucionais baseados na distribuição dos pesos de uma camada e nos valores do fator de normalização de camadas *Batch-Normalization*, respectivamente. Ambos artigos demonstram resultados em diferentes redes com redução de aproximadamente 60% do número de parâmetros na rede sem perda de acurácia.

A maioria dos trabalhos que focam na compressão por poda, buscam uma relação entre pesos e importância de parâmetros em uma rede. Por exemplo, os autores Jordao, Yamada e Schwartz (2020), utilizam o método de regressão por mínimos quadrados parciais para representação dos filtros em um hiperespaço, o qual é utilizado para demonstrar a importância de cada filtro convolucional durante a inferência da rede. Apesar do elevado custo computacional, o trabalho demonstra a compressão de uma rede em 60% com aumento considerável de sua acurácia.

Para a quantização de redes convolucionais em (ZHOU et al., 2017) é apresentada uma abordagem onde os pesos da rede são quantizados para valores 2^n, sendo n um número inteiro e finito. O artigo demonstrou resultados de redes 89x menores sem uma perda considerável na acurácia, e reduções de 2% para quantização binária da rede.

Já em Jacob et al. (2017), os autores abordam um método para quantização dos parâmetros da rede para valores inteiros em 8bits, os resultados demonstram uma redução em 4x no custo de memória e uma queda pela metade no tempo de inferência.

Porém muito dos trabalhos vistos na literatura possuem uma implementação inacessível, ou pouco flexíveis onde o algoritmo funciona para as redes e configurações citadas no trabalho.

Por outro lado há também bibliotecas de código aberto que possuem uma abordagem mais genérica, onde o algoritmo é versátil o suficiente para o uso de diferentes hiper-parâmetros e RNAs. O TensorFlow (ABADI et al., 2015), sendo uma biblioteca de aprendizado de máquina, oferece implementações de código para a compressão de redes neurais artificiais.

Além do TensorFlow disponibilizar três métodos de compressão, sendo a poda de pesos, quantização para 8 ou 16 bits e compartilhamento de pesos, a biblioteca da a liberdade para a definição dos hiper-parâmetros de configuração para aplicação dos métodos.

Entretanto o que se observa na literatura e na indústria de software é um avanço considerável nos métodos de compressão de RNAs, porém sem uma discussão sobre o impacto dos hiper-parâmetros de compressão nas características da rede comprimida.

Assim neste trabalho propõe-se um algoritmo de otimização para a escolha automática dos hiper-parâmetros de compressão implementados pela biblioteca TensorFlow.

O algoritmo desenvolvido se baseia na mesma premissa utilizada em outros trabalhos sobre compressão de RNA, que é a proposta de um método que possibilite a redução do custo computacional de uma RNA com o menor impacto negativo na eficácia da rede.

Com esta premissa, modelamos um problema bi-objetivo de compressão de RNA, onde

buscamos incrementar a acurácia da rede e reduzir o armazenamento da rede, que é um indicador simples para correlação com custo computacional.

A compressão de uma rede entretanto é demorada, e pode tornar o uso de métodos de otimização direta inviável, visto que estes métodos geralmente necessitam de muitas avaliações de soluções para resultados satisfatórios. Dado esta restrição de número de avaliações da função de compressão, é utilizada uma abordagem de otimização através de modelos substitutos, que se mostra como uma ótima solução para otimização de problemas com alto custo para avaliação.

Para a avaliação do algoritmo proposto, foram feitos testes de compressão das redes ResNet50 e VGG16 e os resultados comparados com as soluções encontradas utilizando como otimizador o algoritmo evolutivo *NSGA-II* (Deb et al., 2002), onde as restrições para avaliação da função de compressão se mantiveram iguais.

1.1 Objetivos gerais e específicos

O objetivo principal deste projeto é implementar e analisar métodos de compressão de RNA já propostos em outros trabalhos e propor uma metodologia para definir automaticamente a estratégia de compressão mais adequada.

Este objetivo principal pode ser dividido nos seguintes objetivos específicos:

- Estudar os métodos de compressão propostos na literatura;
- Automatizar a escolha de parâmetros dos métodos mais promissores;
- Investigar diferentes configurações de parâmetros e taxas de compressão para os métodos;
- Investigar a combinação de diferentes métodos;
- Testar a metodologia proposta em diferentes arquiteturas de redes neurais artificiais.

1.2 Organização do texto

Esta monografia segue a seguinte organização de capítulos:

- No capítulo 2 serão apresentados conceitos base para os métodos de compressão utilizados;
- O capítulo 3 trata de conceitos de otimização multiobjetivo e otimização por modelos substitutos que serão utilizados no algoritmo proposto;
- No capítulo 4 é dedicado a metodologia, onde é apresentado a formulação formal do problema e da solução investigada;
- O capítulo 5 apresenta os resultados dos testes propostos no capítulo anterior;

- O capítulo 6 finaliza o trabalho com as conclusões retiradas dos resultados e possíveis caminhos para futuras pesquisas

2 REDES NEURAIS ARTIFICIAIS E MÉTODOS DE COMPRESSÃO

2.1 Redes Neurais Artificiais

Redes Neurais Artificiais (RNA) são algoritmos de aprendizado de máquina utilizados para a aproximação de funções matematicamente complexas (DEEP..., 2021). Assim como um gráfico cheio de curvas pode ser aproximado por séries de Fourier (OSGOOD,), sendo estas nada mais que uma soma finita de senoides, uma RNA utiliza um conjunto de funções simples, representadas por neurônios, para aproximar uma função matemática. Em outros termos, redes neurais são grandes funções compostas, formadas por funções menores, geralmente caracterizadas por neurônios matemáticos ou nós.

O neurônio matemático é composto por alguns elementos, como representado na Figura 1, sendo estes os sinais de entrada x_i, os pesos de cada entrada w_i, o viés θ, e a função de ativação $g(\cdot)$, e yrepresenta a saída ou resposta do neurônio.

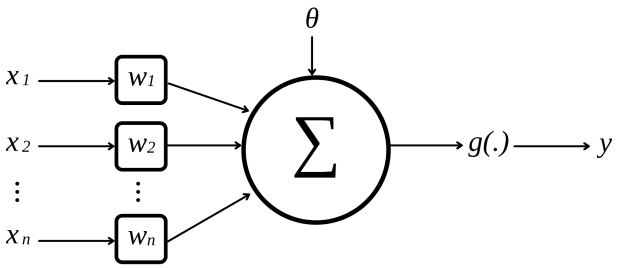

Figura 1 – Representação de um neurônio artificial

A equação 2.1 define a forma genérica que descreve a função de um neurônio artificial.

$$y = g\left(-\theta + \sum_{i=1}^{n} x_i w_i\right) : \text{sinal de saída} \tag{2.1}$$

Uma rede neural artificial é composta por uma grande quantidade de neurônios que são dispostos em camadas. Estes neurônios são conectados entre si, recebendo, em geral, o sinal de saída da camada anterior e alimentando neurônios da camada posterior. Existem diversas formas de se formar conexões em uma rede neural, a forma como as camadas de neurônios se conectam definem a arquitetura da rede. Entre as mais básica está a rede densa demonstrada na Figura 2.

As camadas podem ser divididas em três tipos, camada de entrada, camadas ocultas e camada de saída.

A camada de entrada adquiri os dados de entrada e os repassa para a camada seguinte, essa camada pode por exemplo representar os valores dos pixels de uma imagem. As camadas

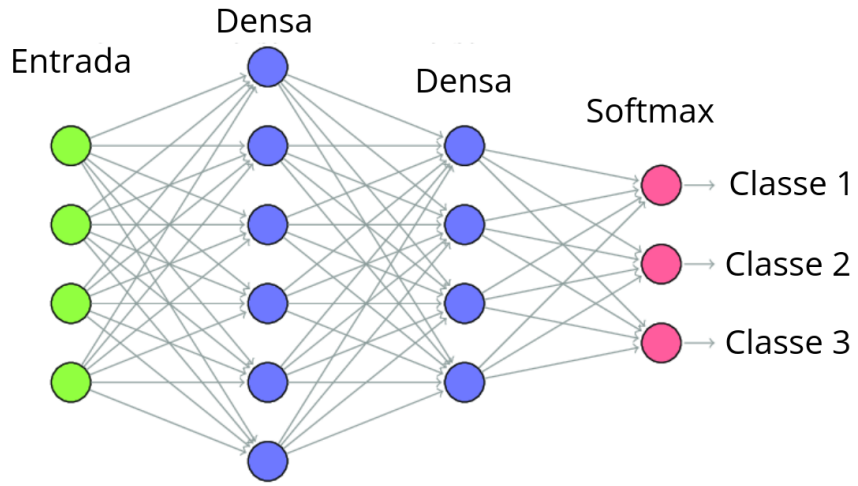

Figura 2 – Representação de um rede neural artificial.
Adaptado de https://www.researchgate.net/figure/Example-of-fully-connected-neural-networZ_fig2_331525817

ocultas são responsáveis por mapear os dados de entrada em padrões úteis. Por exemplo, enquanto uma camada pode identificar bordas, uma camada oculta seguinte pode identificar contornos ou objetos. Por fim, a camada de saída gera o resultado dos cálculos da rede neural. Exemplificando, em um problema de classificação de imagens esta camada pode indicar a probabilidade de uma dada imagem ser de uma determinada classe.

2.1.1 Redes Convolucionais

Redes neurais convolucionais são uma categoria de arquitetura no qual a RNA aplica, em ao menos uma de suas camadas, a operação de convolução ao invés da multiplicação de matrizes observada em camadas densas (DEEP. . . , 2021).

Em uma camada convolucional as conexões de um neurônio só se dão com uma pequena região dos neurônios da camada anterior, chamada de campo receptivo, como mostrado na Figura 3.

Em um camada convolucional os neurônios também compartilham dos mesmos pesos, esta matriz de pesos compartilhada é chamada de filtro ou *kernel*. As equações 2.2 e 2.3 descrevem uma camada convolucional com entrada de 2 dimensões, onde I indica a matriz de entrada e K o filtro da camada. O filtro das camadas convolucionais as permitem identificar um mesmo padrão em diferentes localizações de uma matriz de entrada.

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \text{ : matriz de convolução} \quad (2.2)$$

$$y_{i,j} = g(-\theta + S(i,J)) \text{ : saída de um neurônio de uma camada convolucional} \quad (2.3)$$

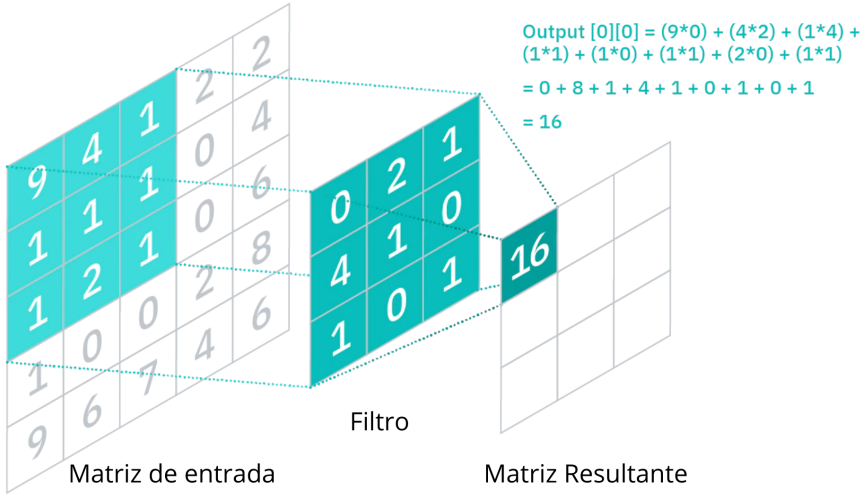

Figura 3 – Representação de uma camada convolucional.
Adaptado de https://www.ibm.com/cloud/learn/convolutional-neural-networks

Além do cálculo da convolução e da função de ativação dos neurônios 2.2 e 2.3, geralmente as camadas convolucionais são seguidas de camadas de *pooling*.

A camada de *pooling* condensa as saídas de uma camada convolucional, gerando uma matriz resultante menor. Uma camada de *pooling*, assim como uma camada convolucional, possui cada unidade conectada à apenas uma região da camada anterior como mostrado na Figura 4. A diferença entre as camadas é que a camada de *pooling* não possui pesos para cada conexão, ela é definida por uma função matemática, geralmente média ou valor máximo.

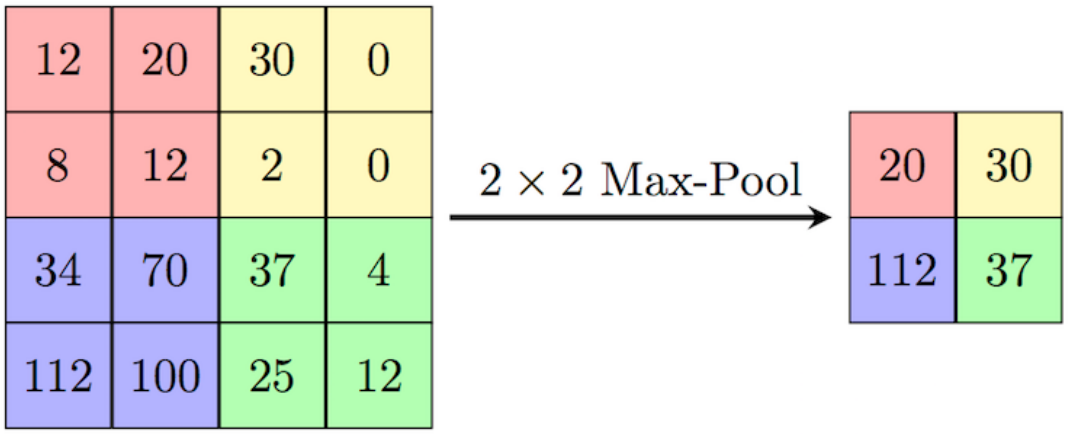

Figura 4 – Representação do funcionamento de camada de *pooling*.
Fonte: https://paperswithcode.com/method/max-pooling

2.1.2 Treinamento de uma rede neural artificial

Para que uma RNA possa aproximar uma função, ela passa por um processo de treinamento supervisionado no qual é necessário um conjunto de dados de entrada, X, com os resultados esperados, $\mathbf{y}$, conjunto chamado de base de dados ou *dataset*.

Durante o treinamento, uma parcela, ou lote, dos dados de entrada do *dataset* são alimentados na rede. Com os resultados gerados pela RNA e com os resultados esperados é possível calcular o erro da rede e atualizar os pesos e vieses com um algoritmo de otimização de forma a minimizar este erro (DEEP. . . , 2021).

Um passo de treinamento acontece quando a rede é alimentada com um lote de dados e tem seus parâmetros atualizados. O conjunto dos n passos necessários para que todos os dados de treinamento sejam utilizados leva o nome de época de treinamento.

Após cada época de treinamento os parâmetros são salvos caso o erro calculado seja menor que o da época anterior. O treinamento é interrompido após uma condição ser satisfeita, e.g. limite do número de épocas. A Figura 5 demonstra as etapas do treinamento de uma RNA.

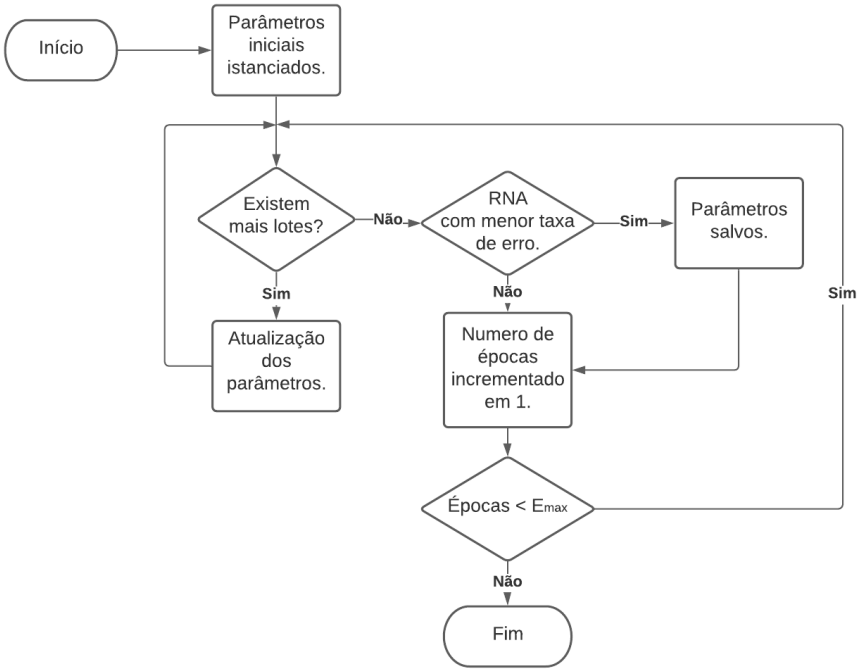

Figura 5 – Fluxograma do treinamento em lote de uma rede neural artificial.

2.2 Métodos de compressão de Redes Neurais

Os métodos de compressão possuem como objetivo reduzir o custo computacional de uma RNA, como o espaço para armazenamento ou latência. Porém, esta compressão pode ter um impacto negativo na eficácia da rede, por isso a aplicação destes métodos deve sempre levar em conta o equilíbrio da eficácia e eficiência da rede. E nesta seção será apresentado os métodos que serão utilizados durante o trabalho.

2.2.1 Poda

A poda de RNAs é um técnica para redução do número de parâmetros de uma rede neural, apresentado a primeira vez por Cun, Denker e Solla (1989). As diversas técnicas de poda podem agir sobre componentes específicos da rede. Os métodos podem anular certos pesos e vieses da rede cortando conexões. Outras abordagens anulam todas as conexões de um neurônio, podando-o da rede e táticas mais agressivas podem até podar camadas inteiras de uma rede.

Entre os métodos de poda, um dos pontos críticos se dá na decisão de como serão escolhidos os parâmetros que devem ser podados. Um dos métodos mais comuns para poda verifica a magnitude dos pesos (ZHU; GUPTA, 2017). Assim, pesos de menor magnitude são anulados até que a porcentagem de esparsidade[1] desejada seja atingida na camada podada. Esta será a estratégia de poda utilizada neste trabalho,implementada no módulo *Model optimization* da biblioteca *TensorFlow* (ABADI et al., 2015)

Apesar da poda poder ser aplicada em um rede treinada, a remoção de pesos pode ter um grande impacto negativo na acurácia. Por isso, a poda é aplicada paralelamente ao treinamento ou re-treino da rede, onde as camadas são podadas gradativamente permitindo que a cada passo de poda a rede possa se recuperar.

Durante o treinamento da rede, a cada Δt_p passos, definido como a frequência de poda, uma parcela da matriz de pesos das camadas é zerada até que se atinga uma esparsidade s_t.

Entre os parâmetros para definição da poda estão:

- Frequência de poda(Δt): Define a quantidade de passos de treinamento entre as execuções de poda na rede;
- Camadas a serem podadas: a poda pode ser aplicada em camadas pré-definidas ou em toda a rede;
- Esparsidade final(s_f): a porcentagem de valores nulos desejado na matriz de pesos ao final do treinamento, pode ser definido por camada ou para a rede;
- Passo de inicio (t_0): passo do treinamento no qual a poda é iniciada;
- Passo final (t_f): passo do treinamento que a poda deve se encerrar, geralmente se opta por finalizar a poda antes do treinamento para que o treino possa reparar os danos da poda; e
- Cronograma: algoritmo que define qual a esparsidade desejada para um determinado passo t da poda.

Com os parâmetros de poda definidos, nas próximas subseções será descrito os cronogramas de poda utilizados.

[1] Esparsidade pode ser definida como a quantidade ou porcentagem de elementos nulos em uma matriz.

2.2.1.1 Cronograma por esparsidade constante

Neste algoritmo a esparsidade de uma determinada camada se eleva em porcentagens constantes a cada passo de poda. A Equação 2.4 define a esparsidade de uma matriz de pesos em um passo de treinamento t onde n é o número de passos de poda dado pela Equação 2.5.

$$s_t = s_f - s_f \left(1 - \frac{t - t_0}{n\Delta t}\right), t \in \{t_0, t_0 + \Delta t, ..., t_0 + n\Delta t\} \tag{2.4}$$

$$n = \left\lfloor \frac{t_f - t_0}{\Delta t} \right\rfloor \tag{2.5}$$

2.2.1.2 Cronograma por Decaimento Polinomial

A esparsidade definida pelo decaimento polinomial é dada pelos mesmos parâmetros do cronograma constante de poda e pela esparsidade, s_i, para a primeira iteração da poda t_0, assim como o grau p do polinômio, definida pela equação 2.6

$$s_t = s_f - (s_i - s_f) \left(1 - \frac{t - t_0}{n\Delta t}\right)^p, t \in \{t_0, t_0 + \Delta t, ..., t_0 + n\Delta t\} \tag{2.6}$$

Diferente do cronograma por esparsidade constante, neste cronograma a cada passo de poda são removidos menos parâmetros do que em relação ao passo anterior.

A Figura 6 exemplifica a evolução da esparsidade de uma rede durante o treinamento com aplicação de ambos os cronogramas de poda.

2.2.2 Quantização

Outra das técnicas para redução do tamanho e latência de um RNA é a quantização. O método consiste na redução do número de bits para representação de pesos, vieses e/ou ativações da rede (JACOB et al., 2017). Usualmente estes parâmetros são representadas por números de ponto flutuante de 32 ou 64 bits, enquanto redes quantizadas utilizam 16bits, inteiros de 8bits ou em alguns casos até menos (ZHOU et al., 2017).

Para este trabalho foi utilizado o método de quantização de pesos e ativações para ponto flutuante de 16 bits implementado na biblioteca *TensorFlow* (ABADI et al., 2015).

2.2.3 O custo da compressão

Além do custo sobre a eficácia da rede, a compressão de uma rede é um processo demorado, podendo levar 10 minutos de execução [2]. Este custo de tempo vinculado aos objetivos de redução da rede com menor impacto na eficácia tornam interessante o uso de otimização

[2] compressão ResNet50. executado com i5-8400, GTX-1050Ti e 16GB RAM

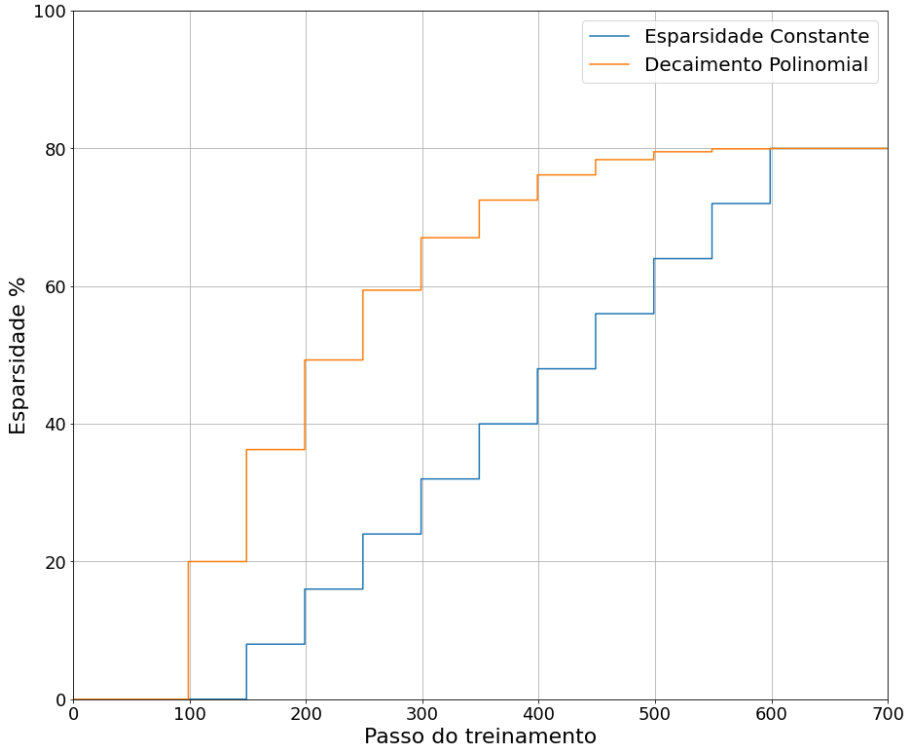

Figura 6 – Esparsidade de uma rede ao longo do treinamento com aplicação do cronograma por esparsidade constante e decaimento polinomial.
Valores utilizados: $s_f = 80$, $s_i = 20$, $\Delta t = 50$, $t_0 = 100$, $t_f = 600$.

multi-objetivo por modelos substitutos, por este tipo de otimização utilizar um número menor de avaliações se comparado com outros otimizadores.

E baseado nas definições dos hiper-parâmetros e custo da compressão, discute-se no próximo capítulo os conceitos base para que possamos criar a metodologia para otimização do problema de compressão.

3 OTIMIZAÇÃO

3.1 Otimização Multiobjetivo

Um problema de otimização multiobjetivo genérico pode ser descrito pela equação seguinte:

$$\begin{aligned} min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x})] \\ s.t. \quad & \mathbf{x} \, \epsilon \, \mathcal{F} \end{aligned} \tag{3.1}$$

Onde $\mathbf{f}$ define o conjunto das m funções objetivo, e $\mathcal{F}$ define a região factível do espaço de busca, ou o conjunto de valores plausíveis de x para o problema.

Como em problemas multi-objetivo assume-se que as funções objetivo são conflitantes entre si, melhorar em uma pode prejudicar outras. Assim, raramente se tem uma única solução.

Para comparação de soluções em problemas multi-objetivo se utiliza do conceito de dominância de Pareto como ilustra a Figura 7. Uma solução $\mathbf{x}_1$ domina $\mathbf{x}_2$ ($\mathbf{x}_1 \preceq \mathbf{x}_2$) se as seguintes condições são satisfeitas:

$$\begin{aligned} &\textit{Para toda função objetivo} \, f_i : f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) \\ &\textit{Ao menos uma função objetivo} \, f_j : f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2) \end{aligned} \tag{3.2}$$

Se tanto $\mathbf{x}_1 \preceq \mathbf{x}_2$ quanto $\mathbf{x}_2 \preceq \mathbf{x}_1$ forem falsas, se diz que são soluções não dominadas.

Como é mostrado na Figura 7, os pontos em verde são pontos não dominados pois qualquer outra solução possui um valor maior em uma das funções objetivos. Enquanto os pontos em azul são soluções dominadas pois existem soluções com valores menores para todas as funções objetivo.

Geralmente, o objetivo dos algoritmos de otimização multi-objetivo é encontrar o conjunto das melhores soluções não dominadas, ou conjunto ótimo de Pareto. O conjunto ótimo de Pareto contém todas soluções não dominadas na região factível onde nenhuma outra solução pode reduzir uma função objetivo sem incrementar outra (SILVA, 2018).

O mapeamento deste conjunto no espaço de objetivos leva o nome de fronteira de Pareto, representada na Figura 8.

3.1.1 Hiper-Volume

Uma das métricas para comparação dos conjuntos ótimos de diferentes algoritmos é o hiper-volume, descrito pela equação 3.3 (SILVA, 2018). Sendo $A = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n\}$ um conjunto

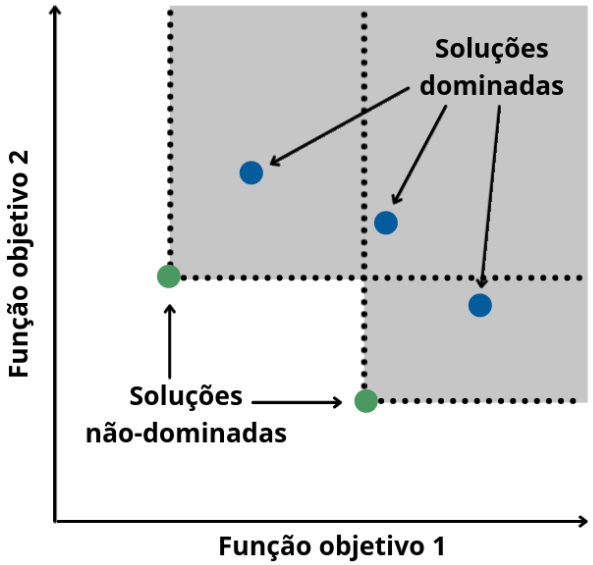

Figura 7 – Representação da relação de dominância de Pareto.
Adaptado de Silva (2018).

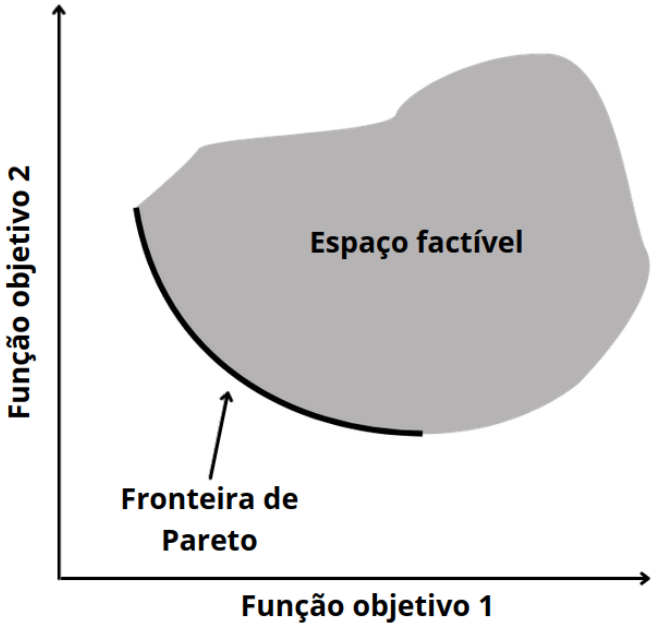

Figura 8 – Representação da fronteira de Pareto.
Adaptado de Silva (2018).

de soluções não dominadas e $\mathbf{z} = \{z_1, z_2, ..., z_m\}$ um ponto de referência no espaço de objetivos. A Figura 9 demonstra o hiper-volume de um problema de dois objetivos.

$$I_{HV}(A, z) = \sum_{i}^{n} \int_{\mathbf{a}_i}^{\mathbf{z}} \alpha(y, \mathbf{a}_i)\, dy$$

Onde:

$$\alpha(y, \mathbf{a}_i) = \begin{cases} 1 & \textit{se } \mathbf{a}_i \preceq y \\ 0 & \textit{ao contrário} \end{cases} \tag{3.3}$$

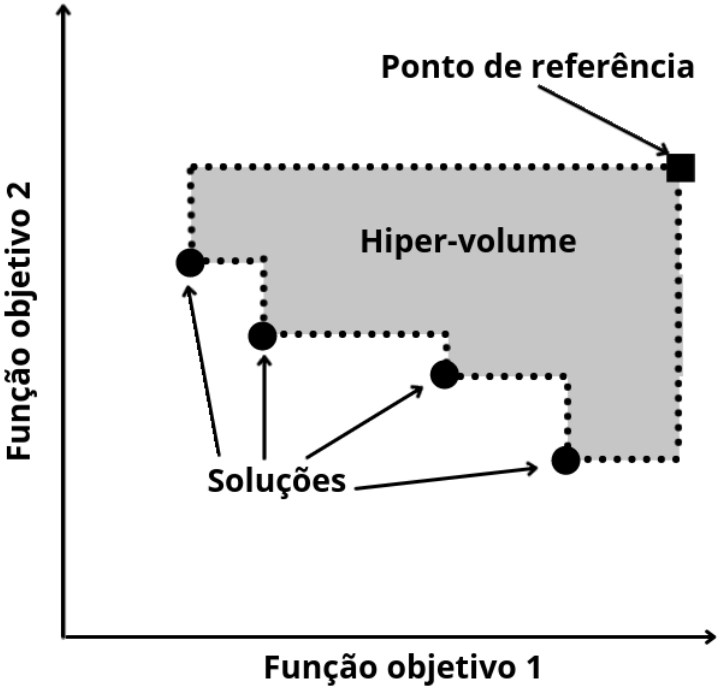

Figura 9 – Hiper-volume de um problema biobjetivo.
Adaptado de Silva (2018).

3.2 Problema de otimização modelos substitutos

Em alguns problemas de otimização a avaliação do modelo real estudado pode ser computacionalmente cara. O método de otimização por modelos substitutos é uma técnica que visa construir um modelo substituto que possa representar o problema original a um custo. O proposito do modelo substituto é facilitar a avaliação do problema ao custo de um acréscimo no erro da representação do problema.

Sendo R_f o modelo original utilizado para analise de um problema de otimização, e R_s um modelo análogo a R_f. Um problema de otimização por modelos substitutos visa resolver indiretamente o problema definido pela equação 3.4 através da equação 3.5.

$$\begin{aligned} &min_(\mathbf{x}) \quad f(\mathbf{x}, R_f(\mathbf{x})) = [f_1(R_f(\mathbf{x})), f_2(R_f(\mathbf{x})), ..., f_i(R_f(\mathbf{x})), f_{i+1}(\mathbf{x}), ..., f_m(\mathbf{x})] \\ &s.t. \quad \mathbf{x}\, \epsilon\, \mathcal{F} \end{aligned} \tag{3.4}$$

$$\begin{aligned} min_{\mathbf{x}} \quad & f_s(\mathbf{x}, R_s(\mathbf{x})) = [f_{s_1}(R_s(\mathbf{x})), f_{s_2}(R_s(\mathbf{x})), ..., f_{s_i}(R_s(\mathbf{x})), f_{s_{i+1}}(\mathbf{x}), ..., f_{s_m}(\mathbf{x})] \\ s.t. \quad & \mathbf{x} \, \epsilon \, \mathcal{F}_s \end{aligned} \tag{3.5}$$

Outra abordagem, utilizada neste trabalho, para modelagem do problema substituto é criar um modelo para representação de cada função objetivo, assim o problema pode ser definido pela equação 3.6. Onde R_{s_i} define um modelo que aproxima o comportamento da função f_i.

$$\begin{aligned} min_{\mathbf{x}} \quad & f_s(\mathbf{x}, R_s(\mathbf{x})) = [R_{s_1}(\mathbf{x}), R_{s_2}(\mathbf{x}), ..., R_{s_i}(\mathbf{x})] \\ s.t. \quad & \mathbf{x} \, \epsilon \, \mathcal{F}_s \end{aligned} \tag{3.6}$$

3.2.1 Algoritmo de otimização por modelos substitutos

O algoritmo utilizado neste trabalho, baseado no trabalho de Silva et al. (2017), consiste nos seguintes passos:

1. Amostragem randômica de um conjunto A de soluções a partir do problema original;
2. Ajuste de n modelos com o conjunto A para representação de cada função objetivo f_i;
3. Seleção do modelo R_{s_i} a partir de uma métrica;
4. Otimização do problema substituto através do algoritmo NSGAII (Deb et al., 2002);
5. Seleção de k *infill points*, ou soluções, geradas no passo anterior;
6. Adição das soluções ao conjunto A;
7. repete o passo 2 até que uma condição de parada seja cumprida;
8. Retorna a fronteira de Pareto obtida do conjunto A.

3.2.2 Métodos de seleção de *infill points*

Como a otimização do problema substituto pode gerar inúmeras soluções, avaliar todas no modelo original pode-se mostrar inviavel, por isso existem alguns métodos para seleção de k soluções que serão avaliadas e adicionadas ao conjunto A de amostras avaliadas. Algumas delas, descritas por Silva et al. (2017) são comentadas a seguir.

3.2.2.1 Seleção Aleatória (RD)

Os k *infill points* são selecionados de forma aleatória do conjunto de soluções gerado pelo algoritmo.

3.2.2.2 Distância no espaço de busca (SD)

Sendo $PS = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$ as soluções geradas pelo algoritmo de otimização e $A = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_m]$ o conjunto de soluções avaliadas no modelo original. O método visa encontrar $\mathbf{x}_i$ o qual minimize a equação:

$$\begin{aligned} & min_{\mathbf{x}_i} \parallel \mathbf{x}_i - \mathbf{a}_j \parallel_2 \\ & j \in \{1, ..., m\} \end{aligned} \tag{3.7}$$

Se k>1, $\mathbf{x}_i$ é adicionado ao conjunto A e a equação 3.7 é reavaliada.

3.2.2.3 Densidade no espaço de objetivos (OD)

Considerando $A_{PF} = [\mathbf{f}_1, ..., \mathbf{f}_m]$ o conjunto que compõe a fronteira de pareto do conjunto A, o infill point selecionado é aquele que minimiza a equação

$$\begin{aligned} & min_{\mathbf{x}_i} \parallel \mathbf{f}_s(\mathbf{x}_i) - \mathbf{f}_j \parallel_2 \\ & j \in \{1, ..., m\} \\ & s.t \ \ \mathbf{x}_i \textit{ não é dominado por } \mathbf{a}_j \end{aligned} \tag{3.8}$$

Caso não haja soluções o suficiente que satisfaçam a equação 3.8, os *infill points* restantes são selecionados aleatoriamente.

No próximo capítulo apresenta-se formalmente a modelagem do problema multi-objetivo de compressão, utilizando os conceitos apresentados na seção 2.2 e neste capítulo.

4 METODOLOGIA

Nesta seção apresenta-se a modelagem de um problema de compressão de redes neurais, a modelagem do algoritmo utilizado para otimização de problemas multiobjetivos por modelos substitutos, e por fim os experimentos realizados.

4.1 O problema de compressão de Redes Neurais Artificiais

O problema de compressão, pode ser considerado um problema multiobjetivo, onde se deseja reduzir o armazenamento da rede e maximar sua acurácia. O problema pode ser descrito pela Equação 4.1.

$$\begin{aligned} &min_{\mathbf{x}} \left\{ f_1(\mathbf{x}) : \text{armazenamento necessário} \right. \\ &max_{\mathbf{x}} \left\{ f_2(\mathbf{x}) : \text{acurácia da rede} \right. \end{aligned} \tag{4.1}$$

Como a maioria dos algoritmos de otimização tratam os problemas de otimização como problemas de minimização a Equação (4.1) pode ser reescrita para:

$$\begin{aligned} &min_{\mathbf{x}} \left\{ f_1(m(M, D, \mathbf{x})) : \text{armazenamento necessário} \right. \\ &min_{\mathbf{x}} \left\{ f_2(m(M, D, \mathbf{x})) : \text{acurácia da rede negativada} \right. \end{aligned} \tag{4.2}$$

A modelagem acima 4.2 apresenta um problema, a função de armazenamento apresenta uma escala de valores muito superior a acurácia, que está entre 0 e 1. Essa diferença de escalas pode ser prejudicial a avaliação de soluções pelo algoritmo de otimização. Por isso equiparamos as escalas das funções objetivos normalizando a função de armazenamento.

O processo de normalização de (f_1) é simples, apenas dividimos o armazenamento necessário da rede comprimida pelo armazenamento da rede original, assim obtemos um coeficiente de compressão da rede, ou coeficiente de peso. Assim o problema de compressão pode ser reformulado para a Equação 4.3.

$$\begin{aligned} &min_{X} \left\{ f_1(\mathbf{x}) : \text{coeficiente de peso} \right. \\ &min_{X} \left\{ f_2(\mathbf{x}) : \text{acurácia da rede negativada} \right. \end{aligned} \tag{4.3}$$

Onde m é a função que descreve a compressão do modelo original (M), D indica o *dataset* utilizado para o treinamento da rede e $\mathbf{x}$ representa um vetor das variáveis de projeto que descrevem os parâmetros de compressão, descritos na seção 2.2 e indicados na tabela 1.

Variável	Definição	Restrições
x_0	Execução da poda	$x_0 = 0$ ou $x_0 = 1$;
x_1	Execução da quantização	$x_1 = 0$ ou $x_1 = 1$
$\mathbf{x}_{2,1...n}$	Tipos de camadas que serão podados	$x_{2,i} = 0$ ou $x_{2,i} = 1$;
$\mathbf{x}_{3,1...n}$	Esparsidade final para cada tipo de camada	$0 \leq x_{3,i} \leq 1$
x_4	Cronograma utilizado para poda	$x_4 = 0$ ou $x_4 = 1$
x_5	Frequência de poda	$x_5 \in Z$ e $x_5 > 0$

Tabela 1 – Variáveis de projeto.

Das variáveis de projeto, x_0 e x_1 são variáveis binárias que definem a aplicação, ou não, do respectivo método de compressão(x_0 para poda e x_1 para quantização). $\mathbf{x}_2$ é um vetor de variáveis, de comprimento igual ao número de tipos de camadas que o modelo M possui, como camadas convolucionais ou densas, onde cada posição indica a aplicação ou não da poda naquele respectivo tipo de camada. $\mathbf{x}_3$ por sua vez é um vetor de tamanho igual à $\mathbf{x}_2$, e armazena valores reais entre 0 e 1, indicando a porcentagem de esparsidade final para cada tipo de camada.

As variáveis x_4 e x_5 são referentes ao processo de poda, onde x_4 é uma variável binária que indica o tipo de cronograma utilizado para a poda da rede, onde 0 indica o uso de decaimento constante e 1 decaimento polinomial. Enquanto x_5 possui um valor inteiro positivo referente a frequência de poda.

4.2 Implementação do Algoritmo Proposto

O algoritmo implementado[1] pode ser divido em dois componentes principais, o problema de compressão e o otimizador por modelos substitutos. Essa divisão permite a independência entre as partes, possibilitando a troca do problema ou otimizador sem que acha um impacto no funcionamento.

4.2.1 Implementação do problema de compressão

O componente que abstrai o problema utiliza a representação de problemas multi-objetivos disponibilizada na biblioteca pymoo (Blank; Deb, 2020)[2]. Para as funções de compressão e treinamento da rede discutidas no capítulo 2, o componente ainda estende algumas funções da biblioteca TensoFlow (ABADI et al., 2015).

Assim o componente é capaz de, a partir de um vetor $\mathbf{x}$, criar uma rede comprimida e gerar os valores das funções objetivo, para qualquer RNA, M e *dataset*, D, como definido na Eq. 4.2.

[1] https://github.com/rcpsilva/EANNCompress

[2] Pymoo é uma biblioteca de código aberto com interfaces e implementações de algoritmos de otimização

4.2.2 Implementação do otimizador por modelos substitutos

O otimizador nada mais é que a implementação do algoritmo discutido na seção 3.2.1. Para escolha do melhor modelo substituto foi utilizado como métrica o erro quadrático médio (MSE) definido pela Equação 4.4. Os modelos disponíveis para serem utilizados durante a otimização como modelos substitutos são os modelos de regressão implementados pela biblioteca Scikit-learn (PEDREGOSA et al., 2011).

$$MSE = \frac{\sum(y_i - \hat{y}_i)^2}{n} \tag{4.4}$$

Onde y é o valor real, $\hat{y}$ o valor medido e $\bar{y}$ a média dos valores medidos.

4.3 Experimentos Computacionais

A avaliação do algoritmo proposto foi feito em duas etapas:

1. Na primeira etapa foram utilizados problemas multiobjetivo de *benchmark* por Zitzler, Deb e Thiele (2000) para a avaliação da robustez e eficiência do método escolhido. Nessa etapa foi avaliado o resultado do método de otimização para diferentes problemas. Os problemas escolhidos são multiobjetivos e famosos na literatura, permitindo a comparação de resultados com outros trabalhos, e também possuem implementações em código-livre (Blank; Deb, 2020).

2. A segunda etapa foi avaliado o algoritmo em dois problemas de compressão, para isso foram utilizados os modelos ResNet50 e VGG16, treinados no dataset Cifar-10 (KRIZHEVSKY, 2012), os modelos e banco de dados foram escolhidos por serem utilizados em outros trabalhos na área de compressão de RNAs, permitindo assim a comparação de resultados.

4.3.1 Experimento de Benchmark

O primeiro experimento visa testar e comparar a eficiência do algoritmo em problemas de teste variando suas configurações. O experimento é feito sobre as versões 2,3,4 e 6 do problema ZDT (ZITZLER; DEB; THIELE, 2000) que contém 2 objetivos e 10 ou 30 variáveis, se assemelhando ao escopo do problema de compressão.

Para comparação, os testes foram feitos utilizando o algoritmo *NSGA-II* diretamente e otimização por modelos substitutos, os parâmetros de ambos são demonstrados nas Tabelas 2 e 3, respectivamente. Para cada configuração de parâmetros foi calculado o hiper-volume da solução através 10 execuções. Foram feitos dois ensaios comparando os resultados entre uma população menor com mas gerações, em relação a um ensaio com maior população e menos gerações. Foi optado por utilizar 150 avaliações da função original para equipar ao problema de compressão, onde este valor equivale a aproximadamente 48 horas de execução.

Parâmetro	Valor
Tamanho da população	10
Número de Gerações	15

Tabela 2 – Parâmetros utilizados na otimização por *NSGA-II*.

Parâmetro	Valores ensaio *a*	Valores ensaio *b*
número de amostras iniciais	30	30
Método de seleção de *infill points*	RD, SD, OD	RD, SD, OD
Número de *infill points*	2, 5, 10	2, 5, 10
Número de avaliações da função	150	150
Tamanho da população*	200	5
Número de Gerações*	350	1000

Tabela 3 – Parâmetros utilizados para algoritmo de otimização por modelos substitutos. *Parâmetros referentes a otimização do modelo substituto por *NSGA-II*.

4.3.2 Experimento de Compressão

Para o teste do algoritmo em um problema de compressão com um rede neural, foram utilizadas uma rede de arquitetura ResNet50 e outra VGG16. Ambas os modelos treinados para tarefa de classificação de imagens através do *dataset* Cifar-10.

Para o processo de otimização dos parâmetros de compressão foi utilizado 30% dos dados para o re-treino e 10% dos dados para avaliação da rede resultante. Optou-se por utilizar apenas uma parcela dos dados de treinamento para reduzir o custo computacional do re-treino e avaliação durante a compressão, assim permitindo um número maior de avaliações da função de compressão

Como a compressão da rede é uma função computacionalmente cara e demorada, foi definido como critério de parada um número máximo de 150 avaliações da função. Foram utilizados os mesmos parâmetros do experimento anterior para otimização por *NSGA-II*. A Tabela 4 demonstra os parâmetros utilizados para a otimização por modelos substitutos, valores estes baseados nos resultados obtidos a partir dos experimentos de benchmark.

Parâmetro	Valor
número de amostras iniciais	30
Método de seleção de modelo substituto	OD
Número de *infill points*	2
Número de avaliações da função	150
Tamanho da população[1]	5
Número de Gerações	1000

Tabela 4 – Parâmetros utilizados na otimização dos parâmetros de compressão por modelos substitutos
[1]Parâmetros referentes a otimização do modelo substituto.

5 RESULTADOS

5.1 Experimentos de Benchmark

Para a análise dos resultados dos experimentos de benchmark foi utilizado *boxplots* baseado nos valores de hiper-volume calculado, como é discuto na seção 4.3.1. Em síntese, quanto mais próximo a "caixa"está de 1.0 mais próximo está a fronteira de pareto calculada para a real, e quanto menor é sua dimensão menor é a variação dos resultados entre as execuções.

As soluções com uso do *NSGA-II* puro não são demonstradas através dos gráficos, pois em todos os problemas discutidos nenhuma solução encontrada dominou o ponto de referência $[0, 0]^T$. Pelo mesmo motivo, o gráfico para problema ZDT4 também foi omitido.

A Figura 10 ilustra os resultados obtido no experimento para o problema ZDT2, para os experimentos com população de 200 Figura 10.a e para população de 5 Figura 10.b. Podemos ver que independentemente do número de gerações e população, em ambos os experimentos as soluções sofrem um grande impacto no uso do método de seleção por distância no espaço de busca, impacto que é minimizado com o aumento de do número de *infill points*.

Tanto na Figura 10.a quanto Figura 10.b podemos ver que o uso da densidade no espaço de objetivos ou seleção aleatória apresentaram resultados muito próximos do real e inalterados pela mudança de *infill points*. O que se pode perceber comparando os resultado porém é uma menor variação das soluções no experimento com menor população e maior número de gerações.

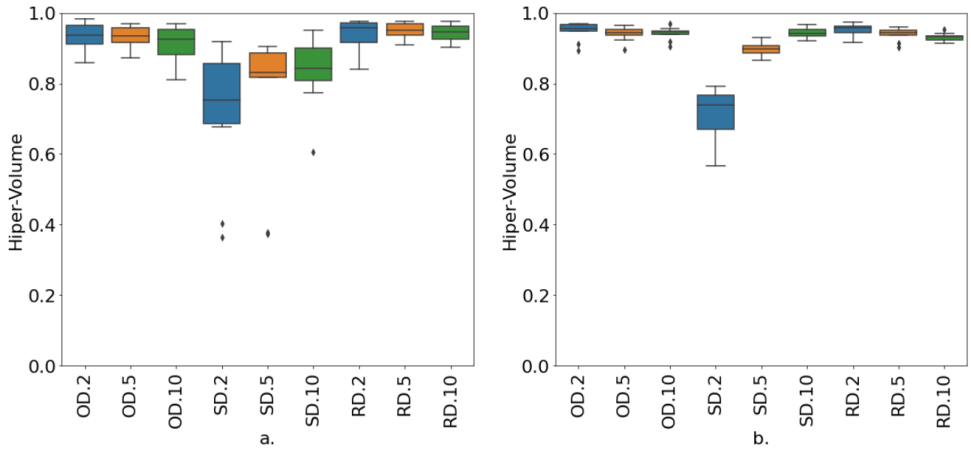

Figura 10 – Diagrama de caixa dos hiper-volumes calculados para solução através de modelos substitutos do problema ZDT2.
a. população 200, número de gerações 350; b. população 5, número de gerações 1000.

Na Figura 11 podemos ver os resultados para o problema ZDT3. O aumento de *infill points* tem um impacto negativo claro nas soluções, indicado principalmente pelos diagramas em verde. Para o ensaio *b* o impacto é tal que quase nenhuma solução factível é encontrada.

Apesar do impacto de muitos *infill points*, para 2 se encontra as melhores soluções. E assim como no experimento passado, densidade no espaço de objetivos se mostra como o melhor método de seleção. Os resultados também demonstram que o ensaio de menor população e mais gerações provê soluções melhores e mais concisas ao longo das execuções.

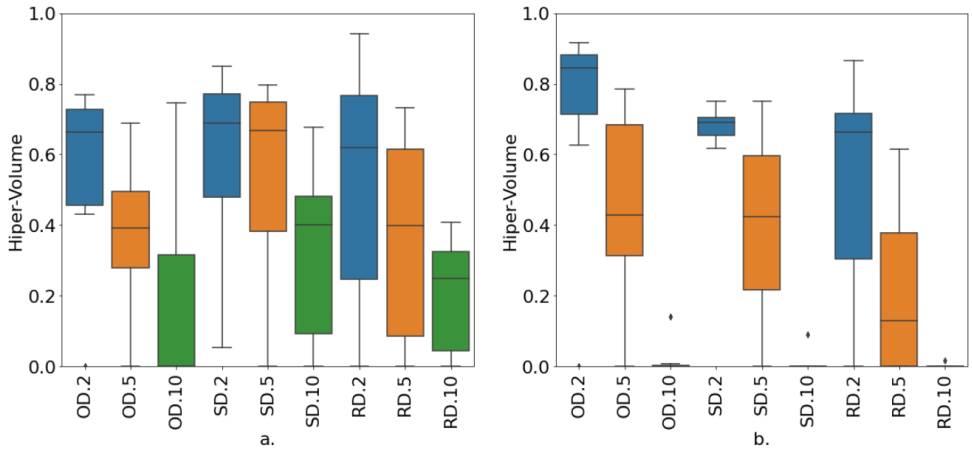

Figura 11 – Diagrama de caixa dos hiper-volumes calculados para solução através de modelos substitutos do problema ZDT3.
a. população 200, número de gerações 350; b. população 5, número de gerações 1000.

A Figura 12 por sua vez indica os resultados dos ensaio *a* e *b* para o problema ZDT6. Neste experimento a vantagem dos parâmetros do ensaio b são indiscutíveis ao se observar que os diagramas da Figura 12.b são todos superiores aos diagramas equivalentes da Figura 12.a. A Figura 12.b demonstra como melhor escolha, o uso de seleção por OD. e de apenas 2 *infill points*.

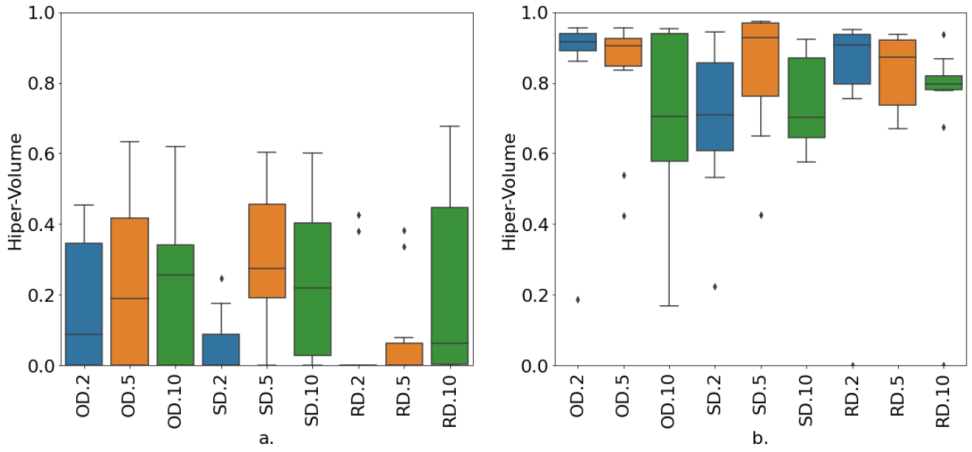

Figura 12 – Diagrama de caixa dos hiper-volumes calculados para solução através de modelos substitutos do problema ZDT6.
a. população 200, número de gerações 350; b. população 5, número de gerações 1000.

Com os resultados obtidos e discutidos nesta seção pode se prosseguir para o experimentos de compressão. E baseado nos resultados desta seção optou-se pelo uso da seleção por

densidade no espaço de objetivo com 2 *infill points* por esta configuração apresentar os melhores resultados e se mostrar concisos durante as execuções nos experimentos de benchmark.

5.2 Experimentos de Compressão de Redes Neurais Artificiais

Esta seção expõe as algumas das conclusões retiradas dos experimentos de compressão da rede Resnet50 e VGG16. Para facilitação da comparação dos resultados entre as redes os resultados serão mostrados em 4 subseções, onde em cada uma será discutido uma visão do resultado.

5.2.1 Fronteira de Pareto

Nesta seção é abordado e comparado os resultados gerados na compressão através do método proposto por modelos substitutos, e *NSGA-II*. Para esta comparação observamos a fronteira de Pareto resultante de ambos os métodos, pois em um problema de otimização multi-objetivo a fronteira é resultado de interesse para a tomada de decisão.

A Figura 13 ilustra as soluções não dominadas encontradas por ambos os métodos de otimização testados para a rede ResNet50. Para ambos os casos as soluções não dominadas se resumem a menos de 10 pontos, porém os pontos gerados pela otimização por modelos substitutos domina a maioria dos pontos gerados diretamente pelo algoritmo *NSGA-II*, excluindo 3 pontos, dois dos quais possuem acurácia menor que 20%, condição que poderia inviabilizar o uso prático da rede.

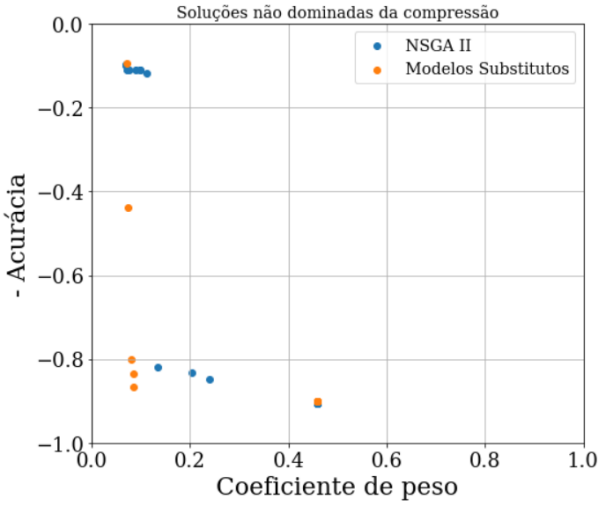

Figura 13 – Fronteira de Pareto para problema de compressão da Rede ResNet50 através de *NSGA-II* e otimização por modelos substitutos.

Assim como para a rede Resnet50, a fronteira de Pareto gerada para compressão da rede VGG16 por modelos substitutos possui menos de 10 pontos (Figura 14). Porém o uso de *NSGA-II* gerou apenas uma solução não dominada para a compressão da VGG16(escondida atrás do ponto laranja em (0.45,-0.9)), ponto gerado quando há apenas a aplicação da quantização na rede.

Como demonstrado em ambas as Figuras 13 e 14, o uso de modelos substitutos para a compressão supera com folga aplicação de *NSGA-II* para um número restrito de avaliações

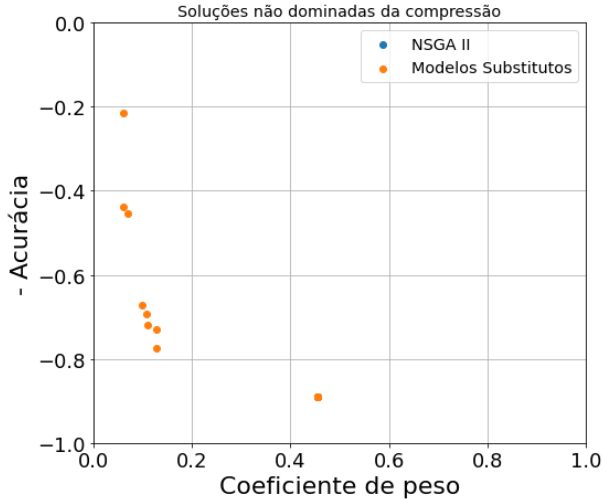

Figura 14 – Fronteira de Pareto para o problema de compressão da Rede VGG16 através de *NSGA-II* e otimização por modelos substitutos.

5.2.2 Soluções por método

Nesta seção e nas seguintes é discutido, não a fronteira de Pareto, mas como os métodos exploram o espaço de variáveis e os impactos destas variáveis na solução.

Para entender o comportamento dos algoritmos em relação as preferencias de aplicação dos métodos de compressão, é demonstrado nesta seção todos os resultados de compressão, dominados e não-dominados, separados pelo método de compressão aplicado.

Na Figura 15 é mostrado o tipo de método aplicado para cada solução gerada na compressão da rede Resnet50, tanto para *NSGA-II*(esquerda) quanto para modelos substitutos(direita). A Figura 15 indica uma vantagem na aplicação simultânea de quantização e poda para compressão da rede. A quantização se mostra lucrativa por reduzir em mais que a metade o coeficiente de peso da rede, com impacto quase nulo na acurácia. A poda por outro lado permite reduzir o peso em faixas livres, porém pode ter um grande impacto, isso pode ser observado ao ver que não existem soluções não dominadas apenas com a poda.

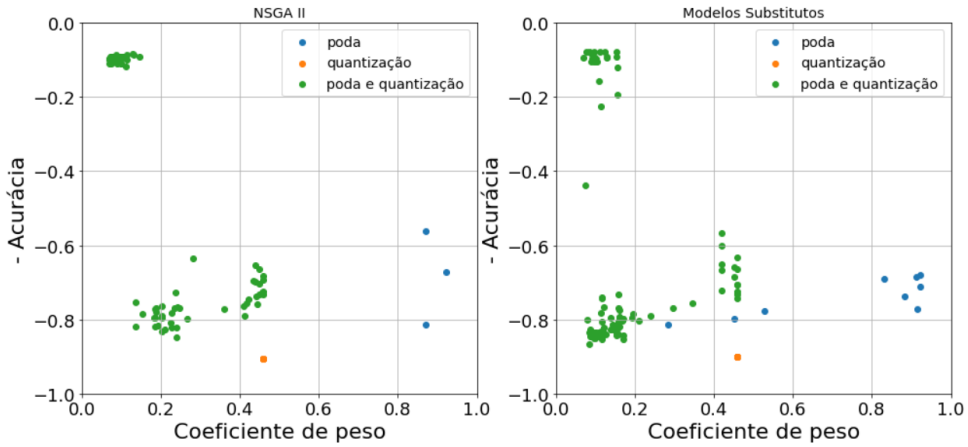

Figura 15 – Soluções de compressão ResNet50 por método aplicado.

Como mostra a Figura 16, diferentemente do que é observado na compressão por *NSGA-II* para ResNet50, este algoritmo repetiu muitas soluções apenas com aplicação de quantização para a compressão da VGG16, não explorando o espaço mais a esquerda do gráfico. Isso indica uma convergência prematura que não ocorreu na versão com modelos substitutos.

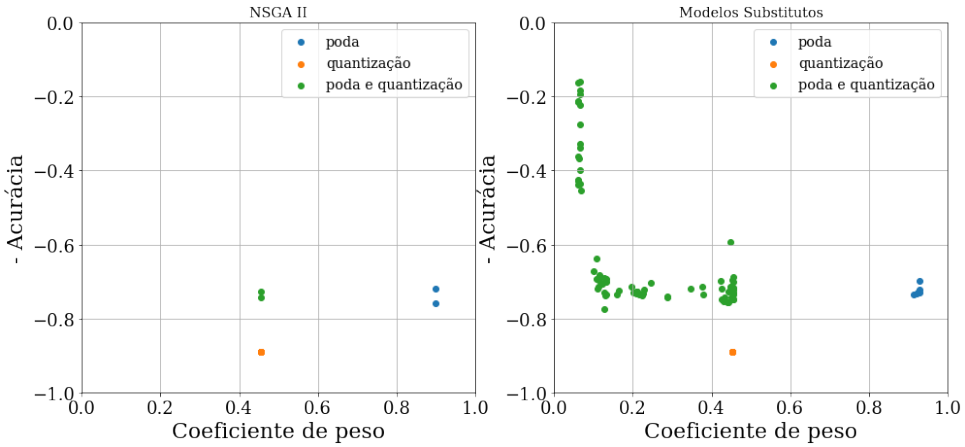

Figura 16 – Soluções de compressão VGG16 por método aplicado.

Apesar da variação de métodos de compressão semelhantes para o uso de *NSGA-II* e modelos substitutos na compressão da ResNet50, observa-se uma grande diferença nos resultados para VGG16. O uso da *NSGA-II* aparenta explorar pouco o espaço travando em um ponto, provavelmente devido ao pouco número de avaliações que o otimizador possui. Por outro lado a otimização por modelos substitutos explora melhor a aplicação de mais métodos devido a sua natureza mais aleatória devido ao processo de constante de otimização do modelo substituto.

5.2.3 Cronogramas de Poda

Nesta seção observa-se apenas a escolha do método de cronograma de poda, logo as soluções com aplicação apenas de quantização são retiradas da análise.

A Figura 17 apresenta as soluções com aplicação de poda do experimento de compressão da rede ResNet50 separadas pelo cronograma de poda. Comparando os resultados dos dois otimizadores, para a compressão da ResNet50, ambos cronogramas atingem reduções do tamanho da rede para até 15% de seu tamanho original com quedas semelhantes de acurácia. Porém, se observa que a partir deste ponto, compressões com uso do decaimento polinomial sofrem um queda brusca de acurácia, contudo a otimização por modelos substitutos atinge soluções melhores que a pura aplicação do *NSGA-II* ao explorar soluções com cronograma de poda constante.

A Figura 18 demonstra os resultados por cronograma para a rede VGG16. Devido à convergência prematura e uso preferencial apenas de quantização, os resultados por *NSGA-II* são inconclusivos. Porém, o uso de modelos substitutos indica que o cronograma utilizado parece alterar em pouco o resultado para compressão da VGG16 como demonstra a Figura 18 à esquerda
.

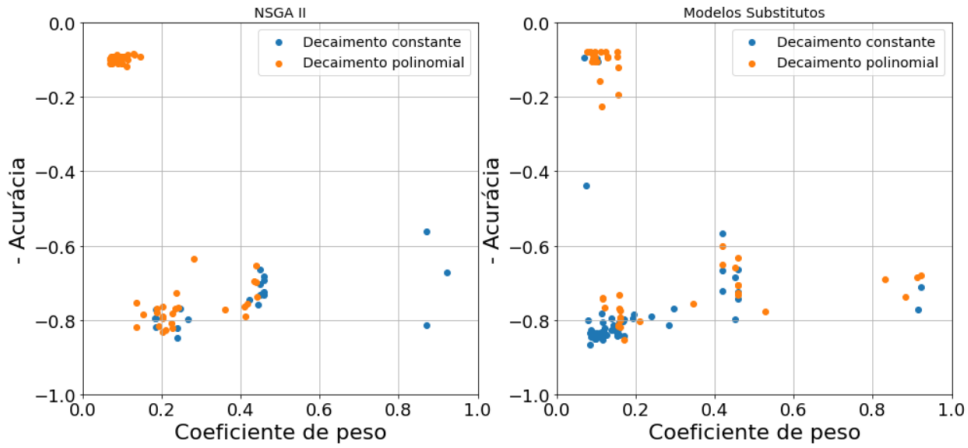

Figura 17 – Soluções de compressão ResNet50 por cronograma de poda.

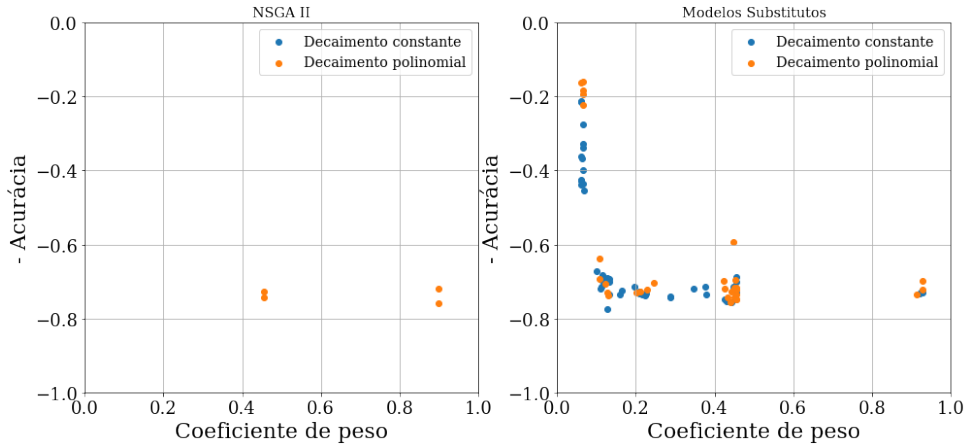

Figura 18 – Soluções de compressão VGG16 por cronograma de poda.

Assim como *NSGA-II* parece convergir para quantização rapidamente na compressão de VGG16, o método parece convergir para o uso de decaimento polinomial em ambas as redes. O método por modelos substitutos, assim como para ResNet50, também explora o uso do decaimento constante.

E comparando os melhores resultados da compressão de ambas as redes, vemos que o decaimento constante é o preferível para compressão da Resnet50, enquanto a VGG16 parece sentir pouca diferença entre os métodos, isso pode indicar que o cronograma ótimo está mais ligado a arquitetura da rede do que a uma constatação genérica para todas as redes.

5.2.4 Frequência de Poda

Nesta seção será analisado o impacto na rede e preferência dos otimizadores sobre a frequência de poda.

A Figura 19 ilustra a frequência de poda baseado em uma escala de cor para cada solução da compressão da rede ResNet50. É possível observar na figura que apesar de soluções semelhantes serem atingidas por frequências abaixo de 100, tanto *NSGA-II* quanto modelos substitutos

demonstraram uma tendência a escolha de frequências de poda baixas para compressão da ResNet50.

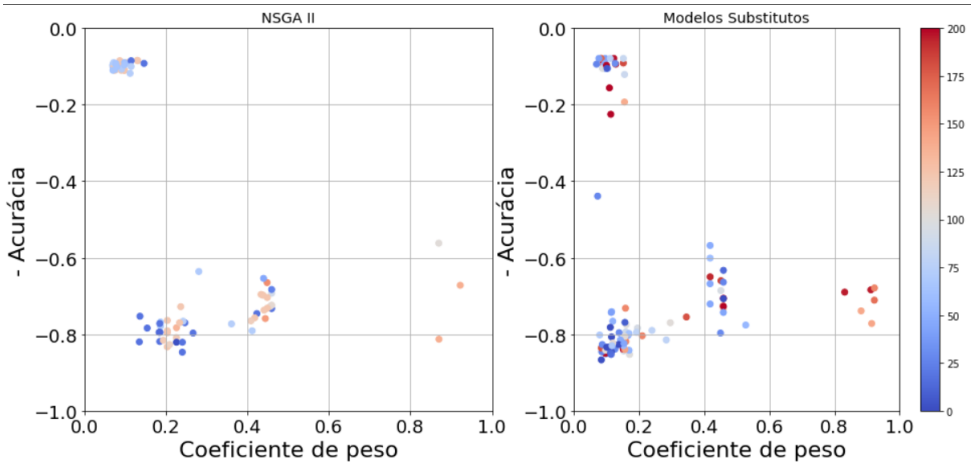

Figura 19 – Soluções de compressão ResNet50 por frequência de poda.

Na Figura 20 as soluções por frequência de poda para VGG16, ao contrário da ResNet50, a preferência das soluções é outro extremo, onde as soluções em sua maioria possuem uma alta frequência.

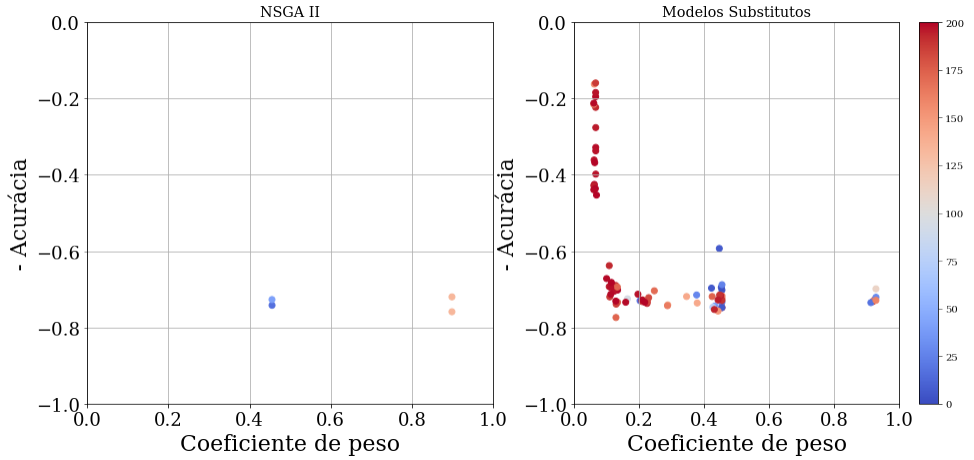

Figura 20 – Soluções de compressão VGG16 por frequência de poda.

Comparando-se os gráficos de compressão da ResNet50, apesar de ambos otimizadores preferirem baixas frequência, observa-se que o uso de modelos substitutos também explora algumas soluções com altas frequências.

Enquanto a comparação da compressão de ambas as redes demonstra a preferência por extremos de frequência opostos, indicando que a frequência ótima também é um parâmetro fortemente vinculado à rede comprimida.

6 CONCLUSÃO

Este trabalho surgiu com intuito de buscar entender os impactos da compressão em uma rede neural artificial. Porque, na área de aprendizado de máquina, onde há esforços constantes visando a melhora dos parâmetros de modelos, sejam RNA ou não, parece ilógico o uso de métodos de compressão de RNA com parâmetros estáticos ou simplesmente aleatórios.

Assim propõe-se um método para otimização destes parâmetros. E o uso de otimização por modelos substitutos se mostrou promissor, obtendo soluções diversas, e melhores que o uso de *NSGA-II* em ambas as redes estudadas.

Porém, apesar da compressão de ambas as redes gerarem resultados significativos, os parâmetros de compressão escolhidos nas melhores soluções se diferem entre as redes, como por exemplo a diferença de frequência de poda. Enquanto na literatura é indicado o uso de altas frequências de poda, os resultados da compressão da ResNet50 demonstraram que o contrário pode ser melhor, enquanto a compressão da VGG16 suporta o uso de altas frequências. Isso pode indicar que valores padrões podem entregar soluções sub-ótimas na maioria dos casos.

O que os resultados e comparações da compressão das redes testadas indicam é que os parâmetros envolvidos na compressão são fortemente vinculados à rede comprimida.

6.1 Trabalhos Futuros

Este trabalho deixa algumas perguntas em aberto que podem ser foco de próximas pesquisas. Como pode-se ver nos resultados diferentes redes possuem diferentes soluções ótimas, logo, poderia se estudar as relações entre a arquitetura de uma rede e os métodos de compressão. Além disso, a formulação do problema ainda pode ser melhorada. Claramente, soluções com 20% acurácia não são viáveis independente do tamanho da rede. Assim, é importante que o método leve em consideração restrições nos objetivos. Levando isso em conta, a modelagem do otimizador ainda precisa ser melhor compreendida no sentido de entendermos quais são os melhores modelos substitutos, se existem maneiras melhores de selecionar *infill points*, e como estes fatores se comportam na presença de restrições. Outro fator dado como fixo neste trabalho, mas que pode ser um foco de próxima pesquisa são os efeitos de outras métricas para seleção de modelos substitutos para o problema de compressão proposto.

REFERÊNCIAS

ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Citado 4 vezes nas páginas 15, 22, 23 e 31.

Blank, J.; Deb, K. Pymoo: Multi-objective optimization in python. *IEEE Access*, 2020. v. 8, p. 89497–89509, 2020. Citado 2 vezes nas páginas 31 e 32.

CUN, Y. L.; DENKER, J. S.; SOLLA, S. A. Optimal brain damage. In: *Proceedings of the 2nd International Conference on Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 1989. (NIPS'89), p. 598–605. Citado na página 22.

Deb, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 2002. v. 6, n. 2, p. 182–197, 2002. Citado 2 vezes nas páginas 16 e 28.

DEEP Learning Book,. [S.l.]: Data Science Academy, 2021. Https://www.deeplearningbook.com.br. Citado 3 vezes nas páginas 18, 19 e 21.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. Http://www.deeplearningbook.org. Citado na página 14.

HE, Y. et al. Pruning filter via geometric median for deep convolutional neural networks acceleration. *CoRR*, 2018. abs/1811.00250, 2018. Citado na página 14.

JACOB, B. et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *CoRR*, 2017. abs/1712.05877, 2017. Citado 2 vezes nas páginas 15 e 23.

JORDAO, A.; YAMADA, F.; SCHWARTZ, W. R. Deep network compression based on partial least squares. *Neurocomputing*, 2020. v. 406, p. 234–243, 2020. ISSN 0925-2312. Citado na página 15.

KRIZHEVSKY, A. Learning multiple layers of features from tiny images. *University of Toronto*, 2012. 05 2012. Citado na página 32.

LIU, Z. et al. Learning efficient convolutional networks through network slimming. *CoRR*, 2017. abs/1708.06519, 2017. Citado na página 14.

LUZ, E. et al. *Towards an Effective and Efficient Deep Learning Model for COVID-19 Patterns Detection in X-ray Images*. 2020. Citado na página 14.

MURPHY, K. P. *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press, 2013. ISBN 9780262018029 0262018020. Citado na página 14.

OSGOOD, B. *The Fourier Transform and its Applications*. [S.l.]: Stanford University. Citado na página 18.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011. v. 12, p. 2825–2830, 2011. Citado na página 32.

SILVA, R. P. *Surrogate Problem Evaluation and Selection for Optimization with Expensive Function Evaluations*. Tese (Doutorado), 05 2018. Citado 3 vezes nas páginas 25, 26 e 27.

SILVA, R. P. et al. Surrogate-based moea/d for electric motor design with scarce function evaluations. *IEEE Transactions on Magnetics*, 2017. PP, p. 1–1, 02 2017. Citado na página 28.

WU, H. et al. *Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation*. 2020. Citado na página 14.

ZHOU, A. et al. Incremental network quantization: Towards lossless cnns with low-precision weights. *CoRR*, 2017. abs/1702.03044, 2017. Citado 2 vezes nas páginas 15 e 23.

ZHU, M.; GUPTA, S. *To prune, or not to prune: exploring the efficacy of pruning for model compression*. 2017. Citado 2 vezes nas páginas 14 e 22.

ZITZLER, E.; DEB, K.; THIELE, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 2000. MIT Press, v. 8, n. 2, p. 173–195, 2000. Citado na página 32.