UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CARLOS DANIEL OLIVEIRA INTERAMINENSE

PODA DE REDES NEURAIS UTILIZANDO O EFEITO CAUSAL ENTRE NEURÔNIOS

CAMPINA GRANDE-PB
2023

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Poda de Redes Neurais Utilizando o Efeito Causal Entre Neurônios

Carlos Daniel Oliveira Interaminense

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Sistemas de Computação

Eanes Torres Pereira (Orientador)

Campina Grande, Paraíba, Brasil

I61p Interaminense, Carlos Daniel Oliveira.
Poda de redes neurais utilizando o efeito causal entre neurônios / Carlos Daniel Oliveira Interaminense. – Campina Grande, 2023.
61 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2023.
"Orientação: Prof. Dr. Eanes Torres Pereira".
Referências.

1. Inteligência Artificial. 2. Podas em Redes Neurais. 3. Efeito Causal. 4. Sistemas de Computação. I. Pereira, Eanes Torres. II. Título.

CDU 004.8(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA SEVERINA SUELI DA SILVA OLIVEIRA CRB-15/225

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO EM CIENCIA DA COMPUTACAO
Rua Aprígio Veloso, 882, Edifício Telmo Silva de Araújo, Bloco CG1, - Bairro Universitário, Campina Grande/PB, CEP 58429-900
Telefone: 2101-1122 - (83) 2101-1123 - (83) 2101-1124
Site: http://computacao.ufcg.edu.br - E-mail: secretaria-copin@computacao.ufcg.edu.br / copin@copin.ufcg.edu.br

FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES

CARLOS DANIEL OLIVEIRA INTERAMINENSE

PODA DE REDES NEURAIS UTILIZANDO O EFEITO CAUSAL ENTRE NEURÔNIOS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação como pré-requisito para obtenção do título de Mestre em Ciência da Computação.

Aprovada em: 14/04/2023

Prof. Dr. EANES TORRES PEREIRA, UFCG, Orientador

Prof. Dr. HERMAN MARTINS GOMES, UFCG, Examinador Interno

Profa. Dra. JOSEANA MACÊDO FECHINE RÉGIS DE ARAÚJO, UFCG, Examinadora Interna

Prof. Dr. FRANCISCO MARCOS DE ASSIS, UFCG, Examinador Externo

Documento assinado eletronicamente por **EANES TORRES PEREIRA**, **PROFESSOR 3 GRAU**, em 17/04/2023, às 10:14, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **HERMAN MARTINS GOMES**, **PROFESSOR 3 GRAU**, em 17/04/2023, às 15:07, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **FRANCISCO MARCOS DE ASSIS**, **PROFESSOR 3 GRAU**, em 17/04/2023, às 15:51, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **JOSEANA MACEDO FECHINE**, **PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 19/04/2023, às 12:57, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

A autenticidade deste documento pode ser conferida no site https://sei.ufcg.edu.br/autenticidade, informando o código verificador **3304467** e o código CRC **7ADA7AEC**.

Referência: Processo nº 23096.027413/2023-10 SEI nº 3304467

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO EM CIENCIA DA COMPUTACAO
Rua Aprígio Veloso, 882, Edifício Telmo Silva de Araújo, Bloco CG1, - Bairro Universitário, Campina Grande/PB, CEP 58429-900
Telefone: 2101-1122 - (83) 2101-1123 - (83) 2101-1124
Site: http://computacao.ufcg.edu.br - E-mail: secretaria-copin@computacao.ufcg.edu.br / copin@copin.ufcg.edu.br

REGISTRO DE PRESENÇA E ASSINATURAS

ATA Nº 003/2023 (DISSERTAÇÃO N° 712)

Aos quatorze (14) dias do mês de abril do ano de dois mil e vinte e três (2023), às quatorze horas (14:00), de forma REMOTA, reuniu-se a Comissão Examinadora composta pelos Professores EANES TORRES PEREIRA, Dr., UFCG, Orientador, funcionando neste ato como Presidente, HERMAN MARTINS GOMES, Dr., UFCG, JOSEANA MACÊDO FECHINE RÉGIS DE ARAÚJO, Dra., UFCG, FRANCISCO MARCOS DE ASSIS, Dr., UFCG. Constituída a mencionada Comissão Examinadora pela Portaria Nº 007/2023 da Coordenadora do Programa de Pós-Graduação em Ciência da Computação, tendo em vista a deliberação do Colegiado do Curso, tomada em reunião de 13 de abril de 2023 e com fundamento no Regulamento Geral dos Cursos de Pós-Graduação da Universidade Federal de Campina Grande - UFCG, juntamente com o Sr(a) CARLOS DANIEL OLIVEIRA INTERAMINENSE, candidato(a) ao grau de MESTRE em Ciência da Computação, presentes professores, alunos do referido centro e demais presentes. Abertos os trabalhos, o(a) Senhor(a) Presidente da Comissão Examinadora anunciou que a reunião tinha por finalidade a apresentação e julgamento da dissertação "PODA DE REDES NEURAIS UTILIZANDO O EFEITO CAUSAL ENTRE NEURÔNIOS ", elaborada pelo(a) candidato(a) acima designado, sob a orientação do(s) Professor(es) EANES TORRES PEREIRA, com o objetivo de atender as exigências do Regulamento Geral dos Cursos de Pós-Graduação da Universidade Federal de Campina Grande - UFCG. A seguir, concedeu a palavra ao (a) candidato(a), o qual, após salientar a importância do assunto desenvolvido, defendeu o conteúdo da dissertação. Concluída a exposição e defesa do candidato, passou cada membro da Comissão Examinadora a arguir o mestrando sobre os vários aspectos que constituíram o campo de estudo tratado na referida dissertação. Terminados os trabalhos de arguição, o(a) Senhor(a) Presidente da Comissão Examinadora determinou a suspensão da sessão pelo tempo necessário ao julgamento da dissertação. Reunidos, em caráter secreto, no mesmo recinto, os membros da Comissão Examinadora passaram à apreciação da dissertação. Reaberta a sessão, o(a) Presidente da Comissão Examinadora anunciou o resultado do julgamento, tendo assim, o candidato obtido o Conceito APROVADO. A seguir, foi encerrada a sessão e lavrada a presente ata, que vai assinada por mim, Paloma Nascimento Porto, pelos membros da Comissão Examinadora e pelo candidato Campina Grande, 14 de Abril de 2023.

Documento assinado eletronicamente por **EANES TORRES PEREIRA**, **PROFESSOR 3 GRAU**, em 17/04/2023, às 09:53, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **HERMAN MARTINS GOMES**, **PROFESSOR 3 GRAU**, em 17/04/2023, às 15:07, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **FRANCISCO MARCOS DE ASSIS**, **PROFESSOR 3 GRAU**, em 17/04/2023, às 15:51, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Carlos Daniel Oliveira Interaminense**, **Usuário Externo**, em 17/04/2023, às 15:58, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **JOSEANA MACEDO FECHINE**, **PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 19/04/2023, às 12:57, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **PALOMA NASCIMENTO PORTO**, **ASSISTENTE EM ADMINISTRACAO**, em 19/04/2023, às 14:39, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

A autenticidade deste documento pode ser conferida no site https://sei.ufcg.edu.br/autenticidade, informando o código verificador **3304462** e o código CRC **C2138ADA**.

Referência: Processo nº 23096.027413/2023-10 SEI nº 3304462

Resumo

O uso de Redes Neurais Profundas (RNP) para resolver problemas de aprendizagem de máquina se tornou comum a partir de 2012, ano em que o modelo AlexNet venceu o desafio da ImageNet, que exigia a classificação de imagens que exigia a classificação de imagens em um conjunto de mil categorias possíveis. Após essa data, outras RNP mais complexas surgiram, chegando a ter bilhões de parâmetros. Assim, aplicar técnicas de poda se tornou uma forma de reduzir a complexidade de uma RNP, pois essas técnicas têm como objetivo remover parâmetros do modelo de entrada, resultando em um modelo menos complexo e com uma acurácia tão boa quanto a obtida pelo modelo de entrada. Nesse contexto, a presente pesquisa propõe uma técnica de poda estruturada que considera o efeito causal entre os neurônios, para decidir quais serão podados, juntamente com todas as suas conexões. Os resultados obtidos nesta pesquisa mostraram que a técnica proposta resulta em modelos com acurácias superiores superiores a outras técnicas de poda investigadas nesta dissertação de podas investigadas e com tempo e ocupação de espaço em disco melhores que o modelo de entrada.

Abstract

The use of Deep Neural Networks (DNN) to solve machine learning problems became common from 2012, in which the AlexNet model won the ImageNet challenge, which required classifying images into a set of a thousand possible categories. Since then, more complex DNN have emerged, with some having billions of parameters. As a result, applying pruning techniques has become a way to reduce the complexity of a DNN, as these techniques aim to remove input model parameters, resulting in a less complex model with accuracies comparable to those of the input model. In this context, the present research proposes a structured pruning technique that considers the causal effect between neurons to decide which ones will be pruned, along with all their connections. The results obtained in this research show that the proposed technique results in models with higher accuracies compared to other pruning techniques investigated in this dissertation, with better time and disk space occupation than the input model.

Agradecimentos

Em primeiro lugar, agradeço a Deus pela conclusão deste trabalho. Só Ele sabe o tanto que pensei em desistir, quantas vezes eu me questionei se era realmente isso que eu queria, por estar cansado e não conseguir produzir o tanto quanto eu esperava nesta pesquisa. Mas tudo foi como Ele quis.

Agradeço também, pela paciência da minha esposa, porque me aguentar durante esses anos de pesquisa não foi fácil, ainda mais com duas gestações, dois partos, crianças, tudo isso logo no começo do nosso casamento e em meio a uma pandemia. Nossa, nem eu me aguentei, mas ela sempre foi e sempre será minha melhor companhia em momentos como os que passamos juntos ao longo dessa pesquisa. Muito obrigado por tudo, Izabella, Bernardo e Maria Clara, amo muito vocês, minha família.

Aos meus pais, pela oportunidade que me deram com minha educação, pois sem o esforço que sempre fizeram por mim jamais eu teria finalizado o meu mestrado. Muito obrigado!

Aos colegas de laboratório/trabalho, meu muito obrigado por todas as trocas, pelas noites jogando CSGO (minha válvula de escape) e por terem me ajudado a me tornar o profissional e pesquisador que sou hoje. Não irei citar nomes, porque são muitas pessoas que estão envolvidas aqui.

Aos meus amigos que fiz ao longo deste tempo e aos que reforçamos ainda mais nossa amizade, em especial a Giuly, Vinícius, Ricardo e Karol, com quem sempre compartilhei meus desafios da pesquisa (em especial ao momento de escrita deste documento, rsrs) e pelas dicas de Arianne e Caio, que também são pesquisadores.

Aos meus orientadores, Eanes e Herman, meu muito obrigado. Foi uma pesquisa difícil, bastante desafiadora, onde eu pude mergulhar num oceano que eu ainda não tinha navegado, o oceano da Causalidade. A pesquisa foi muito demorada, muitas coisas aconteceram ao longo do caminho, mas os senhores jamais me deixaram de lado e sempre respeitaram o meu momento. Agradeço demais pela oportunidade de avançar nesta pesquisa super interessante e pela paciência dos senhores.

Sumário

Glossário

ACE *Average Causal Effect*.

CGNN *Causal Generative Neural Networks*.

CPU *Central Processing Unit*.

GPU *Graphics Processing Units*.

ILSVRC *ImageNet Large Scale Visual Recognition Challenge*.

MCE Modelo Causal Estrutural.

NPU *Neural Processing Units*.

OBD *Optimal Brain Damage*.

OBS *Optimal Brain Surgeon*.

RELU *Rectified Linear Unit*.

RNA Rede Neural Artificial.

RNC Rede Neural Convolucional.

RNP Rede Neural Profunda.

TPU *Tensor Processing Units*.

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Capítulo 1

Introdução

Diversos problemas abordados pela Aprendizagem de Máquina podem ser resolvidos a partir de Redes Neurais Artificiais (RNA), como detecção de faces (SUN; WU; HOI, 2017), rastreamento de pessoas (XUE et al., 2016), entre outros. Modelos de RNA são cada vez mais comuns devido a suas taxas de acerto serem, em geral, bastante elevadas, considerando os mais diversos problemas. Krizhevsky, Sutskever e Hinton (2012) foram pioneiros no contexto de Redes Neurais Profundas (RNP) quando propuseram uma Rede Neural Convolucional (RNC) com 8 camadas, que foi treinada a partir de milhões de imagens para um problema de classificação com 1000 classes. Desde então, modelos cada vez mais complexos foram propostos para resolver os mais diversos problemas no contexto de Visão Computacional e Processamento de Linguagem Natural, por exemplo.

À medida que os modelos neurais se tornam mais complexos, maior passa a ser o custo computacional para a realização de treinamentos e de uso desses modelos em produção. As Unidades de Processamento Gráfico (*Graphics Processing Units* - GPU) se popularizaram como plataforma de hardware para acelerar o treinamento desses modelos, conseguindo reduzir o tempo de treinamento dos modelos, quando comparados com treinamentos realizados em Unidade Central de Processamento (*Central Processing Unit* - CPU). É importante notar que, além das GPUs, também surgiram plataformas de hardware especializadas para o treinamento de RNP, comumente denominadas de *Tensor Processing Units* (TPU) e *Neural Processing Units* (NPU).

Contudo, para realizar inferências em produção, principalmente em cenários em que se faz necessário o uso de um computador com pouco recurso, a complexidade dos modelos

pode ser um fator limitante para a escolha de RNP na solução de problemas. Assim, o contexto de podas em modelos neurais se torna uma opção para contornar essa limitação, já que seu objetivo é reduzir a complexidade desses modelos.

1.1 Motivação

Pearl, Glymour e Jewell (2016) propuseram o Modelo Causal Estrutural (MCE), que é uma forma para modelar problemas analisando suas variáveis e relacionando-as a um grafo acíclico direcionado. A partir de um MCE, é possível ter um entendimento de quais variáveis causam outras, além de servir como base para o cálculo do efeito causal de uma variável em relação a outra. Sabendo que uma RNA pode ser considerada um grafo acíclico direcionado (XIN YAO, 1999), ao representar uma RNA como um MCE, é possível computar o efeito causal entre neurônios do modelo neural. Neste caso, o efeito causal entre neurônios pode ser utilizado como uma técnica para escolher neurônios a serem podados, visando reduzir a complexidade de RNP.

Frankle e Carbin (2019) e Han, Pool et al. (2015) utilizam a técnica de magnitude dos pesos para a seleção de parâmetros a serem podados. Contudo, essa técnica foi proposta para a remoção de parâmetros e não de neurônios. Assim, uma técnica que seleciona neurônios a serem podados, remove do modelo original o neurônio e todas as conexões que chegam e que saem desse neurônio. Esta é a principal diferença entre a técnica por magnitude dos pesos e uma técnica que escolhe os neurônios a serem podados, como uma técnica que faz uso do valor dos efeitos causais entre os neurônios, como base para a escolha desses neurônios.

1.2 Questão de Pesquisa

Como esta pesquisa objetiva usar a relação causa e efeito entre neurônios, como base da técnica de poda em modelos neurais, a seguinte questão de pesquisa foi levantada: **Conhecendo o valor do efeito causal entre neurônios e as saídas de uma rede neural, dada uma massa de dados rotulados como entrada, como decidir se um neurônio deverá ser podado ou não?**

1.3 Objetivos da Pesquisa

A partir da questão de pesquisa apresentada na Seção 1.2, esta pesquisa tem como objetivo geral propor uma técnica de poda que seleciona neurônios, de camadas completamente conectadas, a serem podados a partir do efeito causal entre um neurônio e os neurônios da camada de saída.

Visando a alcançar o objetivo geral, os seguintes objetivos específicos são definidos:

- Definir uma abordagem para computar o efeito causal entre um neurônio presente em uma camada completamente conectada, com relação à saída;
- Usar os valores dos efeitos causais e escolher neurônios a serem podados do modelo;
- Avaliar experimentalmente a abordagem de poda proposta, comparando os resultados obtidos com: o modelo sem poda; modelos resultantes a partir da poda por Magnitude dos pesos e a partir da escolha aleatória dos neurônios a serem podados.

1.4 Contribuições

A presente pesquisa tem como principal contribuição a proposição e avaliação experimental de uma técnica inédita para a poda estruturada de neurônios de camadas completamente conectadas, a partir da análise da relação causa e efeito entre os neurônios com relação à camada de saída. Além disso, a presente pesquisa contribui com uma comparação da técnica proposta com a técnica por magnitude dos pesos, além da técnica de seleção aleatória de neurônios.

1.5 Organização do Trabalho

O presente documento está organizado da seguinte forma: no Capítulo 2, é apresentada toda a fundamentação teórica necessária para entendimento desta pesquisa; no Capítulo 3, são apresentados os trabalhos relacionados com a presente pesquisa; os materiais e métodos são apresentados no Capítulo 4; todos os resultados obtidos a partir dos experimentos realizados são apresentados no Capítulo 5; e, finalmente, no Capítulo 6, são apresentadas as

considerações finais desta pesquisa, bem como os trabalhos futuros que poderão ser realizados a partir da presente pesquisa.

Capítulo 2

Fundamentação Teórica

Os principais conceitos necessários ao entendimento da presente pesquisa são apresentados neste capítulo. Na Seção 2.1, são apresentados conceitos de Redes Neurais Artificiais. Os principais conceitos sobre poda e tipos de poda são apresentados na Seção 2.2. Por fim, na Seção 2.3, são descritos os conceitos de causalidade.

2.1 Redes Neurais Artificiais (RNA)

Uma RNA pode ser definida como um modelo computacional biologicamente inspirado, que consiste em um conjunto de elementos de processamento, também conhecidos como neurônios, que estão interligados por meio de canais de comunicação, conhecidos como sinapses (XIN YAO, 1999). Estes canais de comunicação são representados, em modelos computacionais, por meio de matrizes de pesos, onde o conhecimento adquirido pelo modelo está armazenado. Assim, uma RNA pode ser descrita como um grafo direcionado (XIN YAO, 1999), em que os neurônios são os nós do grafo e os canais de comunicação são as arestas que conectam os nós. Na Figura 2.1, é apresentado um exemplo de RNA com três camadas, sendo, da esquerda para a direita: uma camada de entrada, uma camada escondida e uma camada de saída. Os neurônios são representados pelos círculos e os canais de comunicação pelas flechas que conectam os neurônios.

As Redes Neurais Profundas (RNP) são um tipo de RNA compreendendo arquiteturas com muitas camadas escondidas e muitos neurônios (LIU, W. et al., 2016) (DENG; YU,

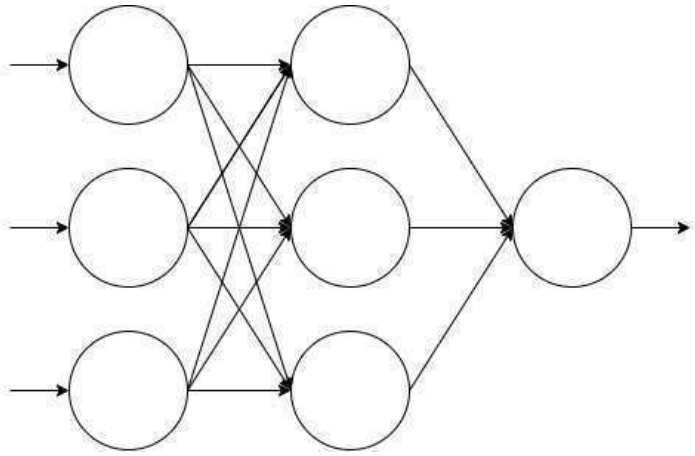


Figura 2.1: Exemplo de uma RNA. Fonte: Autoria própria.

2014). As RNP ficaram famosas a partir de 2012 no desafio da ImageNet[1], cujo objetivo era classificar mais de 100.000 imagens divididas em 1.000 classes, após realizar o treinamento dos modelos a partir de 1,2 milhão de imagens para treino e 50.000 para validação. O desafio daquele ano foi vencido por uma RNP conhecida como AlexNet, que possui mais de 650.000 neurônios e mais de 60 milhões de conexões (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). A AlexNet é um tipo de Rede Neural Convolucional (RNC). As RNC recebem este nome por possuírem em sua arquitetura, camadas convolucionais cujo objetivo é extrair mapas de características e são comumente utilizadas para solução de problemas envolvendo Visão Computacional como os de classificação e segmentação de imagens.

De modo geral, as RNC são compostas por três tipos de camadas: convolucionais, abstração (*Pooling*) e totalmente conectadas (O'SHEA; NASH, 2015). Na Figura 2.2, é apresentado um exemplo de arquitetura de uma RNC, no contexto de classificação de dígitos manuscritos na base de imagens MNIST (DENG, 2012).

Na Figura 2.2, a entrada é composta por uma imagem, que pode ter dimensões padrões, como 224×224 pixels. As camadas convolucionais são responsáveis por extrair mapas de características a partir de operações de convolução de filtros sobre as entradas da camada. Normalmente, após cada camada convolucional, tem-se uma função de ativação, que no caso da Figura 2.2 é usada a *Rectified Linear Unit* (RELU) (NAIR; HINTON, 2010), que computa o valor máximo entre seu valor de entrada e o zero. Em seguida, a camada de *Pooling* é responsável por reduzir a dimensionalidade do mapa de características retornado pela ca-

[1]http://www.image-net.org/challenges/LSVRC/2012/, acessado em 05/05/2020 às 00h48m

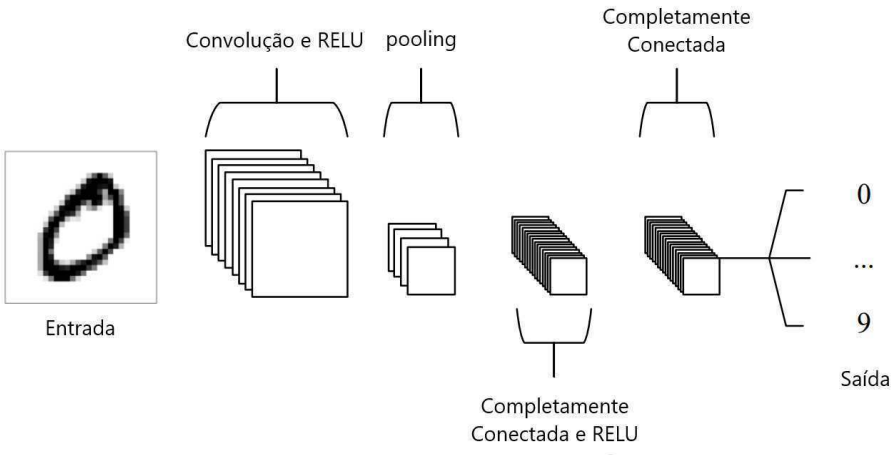


Figura 2.2: Exemplo de uma RNC. Fonte: O'Shea e Nash (2015)

mada convolucional e função de ativação. Finalmente, o mapa de características retornado pela camada de *Pooling* é transformado em um vetor unidimensional e passado como entrada para as camadas completamente conectadas, as quais têm como função computar um *score* e associar os mapas de características a uma classe, definida pela saída da Figura 2.2.

2.2 Poda em Redes Neurais Profundas (RNP)

Sabendo-se que uma RNP pode chegar a ter bilhões de parâmetros, como a RNP GPT-3 (BROWN et al., 2020), técnicas de compressão de modelos neurais podem ser aplicadas. Uma técnica comumente usada é a de poda, cujo objetivo é reduzir a complexidade de modelos neurais por meio da eliminação seletiva de neurônios e/ou conexões, resultando em modelos que consomem menos recursos computacionais (CPU e memória), além de reduzir espaço de armazenamento, buscando manter a acurácia do modelo similar a do modelo original. Tais técnicas de compressão de modelos neurais são utilizadas porque as RNP tipicamente necessitam de capacidades de processamento e armazenamento elevados para realização de inferências, o que pode ser um problema de uso desses modelos em dispositivos com alguma limitação de hardware (HAN; MAO; DALLY, 2016).

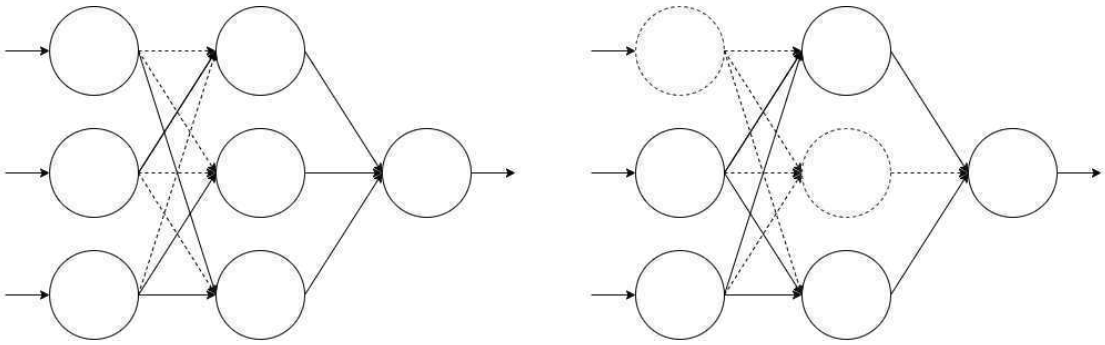
Estratégias de poda em RNP podem ser subdivididas em duas categorias:

i **estruturada**, que considera um grupo de parâmetros, removendo um neurônio e todas as suas conexões, por exemplo. Neste tipo de poda, o resultado é um modelo não esparso,

ou seja, um modelo em que as conexões são de fato removidas e não apenas zerados os seus respectivos pesos. Podas realizadas a partir dessa forma geram modelos menos complexos e com um melhor desempenho, quando comparados aos modelos originais (WANG et al., 2019);

ii **não estruturada**, que considera parâmetros individuais, ou seja, conexões. Conexões podadas recebem 0 (zero) como valor. Assim, o modelo retornado é um modelo esparso, que contém a mesma quantidade de parâmetros do modelo original, mas os pesos das conexões podadas possuem o valor 0 (zero). Em termos de desempenho, modelos podados de forma não estruturada, apenas usando *hardwares* e *softwares* específicos, podem superar o modelo original em relação ao tempo de processamento, memória consumida e até mesmo em acurácia (ZHANG et al., 2019)

A Figura 2.3 apresenta um exemplo de um modelo após a aplicação da técnica de poda não estruturada e da poda estruturada.



(a) Poda não estruturada (b) Poda estruturada

Figura 2.3: Resultado da aplicação de (a) poda não estruturada e de (b) poda estruturada.

Na Figura 2.3, os artefatos tracejados representam as conexões ou neurônios podados a partir do modelo neural apresentado na Figura 2.1, onde: a Figura 2.3a representa o modelo final após a poda não estruturada; e a Figura 2.3b representa o modelo final após a poda estruturada.

Poda por Magnitude dos Pesos

A técnica de poda por Magnitude dos Pesos é bastante utilizada na literatura como nas pesquisas de Han, Pool et al. (2015), Gale, Elsen e Hooker (2019) e Frankle e Carbin (2019).

Um dos motivos da sua popularidade é a velocidade com que ela pode ser aplicada em RNP, já que esta técnica ordena as conexões a partir da magnitude dos pesos (do seu valor absoluto) e remove as conexões que estão abaixo de um determinado limiar (HAN; POOL et al., 2015). Quando esta técnica de poda é aplicada em camadas completamente conectadas, o resultado é um modelo esparso (YEOM et al., 2019).

Poda Aleatória

A técnica de poda aleatória tem como objetivo realizar a escolha dos neurônios ou conexões de forma completamente aleatória (YU et al., 2017). Da mesma forma da poda por magnitude, quando este tipo de poda é aplicado nas conexões o resultado é um modelo esparso. Porém, quando aplicada nos neurônios, o modelo não será um modelo esparso, dado que o neurônio e todas as suas conexões poderão ser removidos.

2.3 Causalidade

Um dos principais objetivos da causalidade é estimar a relação de causa e efeito entre dois eventos (PEARL; GLYMOUR; JEWELL, 2016). Por exemplo, com a causalidade seria possível responder questões do dia a dia, como "Fumar causa câncer de pulmão?", ou "O remédio x é eficaz para o tratamento da doença y?", ou uma questão mais voltada para aprendizagem de máquina usando RNA: "Qual é o efeito causal do neurônio x, da camada i, no neurônio y da camada $i + 1$?".

O entendimento da relação de causa e efeito se mostra importante quando é preciso realizar uma análise de dados considerando uma população ou se a população deve ser fragmentada em subpopulações. A Figura 2.4 apresenta a relação entre a prática de exercícios e taxa de colesterol, agrupando os indivíduos por faixa etária.

A partir da análise da Figura 2.4, percebe-se que existe um comportamento em comum entre as faixas etárias: quanto mais exercício um indivíduo praticar, menor será sua taxa de colesterol. Porém, se o mesmo gráfico da Figura 2.4 for plotado sem considerar a faixa etária, o entendimento é o oposto, como apresentado na Figura 2.5.

A partir da análise da Figura 2.5, percebe-se que existe uma correlação forte e positiva entre as variáveis, levando o leitor ao entendimento de que quanto mais exercício ele pratica,

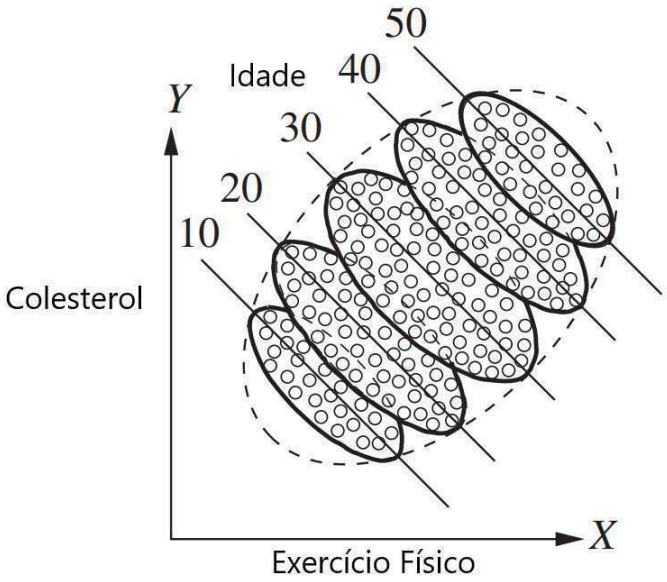


Figura 2.4: Relação entre a prática de exercícios e taxa de colesterol, agrupando os indivíduos por faixa etária. Fonte: Adaptado de Pearl, Glymour e Jewell (2016).

maior será sua taxa de colesterol. Entretanto, é esperado o oposto, ou seja, quanto mais exercício um indivíduo pratica, menor seria sua taxa de colesterol. Mas afinal, qual é a interpretação correta?

Considerando a Figura 2.4, pode-se concluir que pessoas com idades mais avançadas tendem a praticar mais exercícios e ter taxas de colesterol mais elevadas, quando comparadas com pessoas mais jovens. Assim, a variável Idade é causa tanto para Taxa de Colesterol quanto para Prática de Exercícios, ou seja, a avaliação mais correta que deve ser considerada é a análise a partir dos dados agrupados por faixa etária. Em Pearl, Glymour e Jewell (2016), é possível encontrar outros exemplos em que há interpretações opostas ao analisar os dados de forma geral ou agrupada, conhecido como Paradoxo de Simpson.

Uma forma de responder as questões supracitadas, é modelando o problema, ou seja, é preciso identificar variáveis do problema e como elas estão relacionadas. Para isso, Pearl, Glymour e Jewell (2016) propuseram a utilização do Modelo Causal Estrutural (MCE), que é uma forma de descrever as variáveis de um problema e como elas estão relacionadas a partir de um grafo acíclico direcionado. Formalmente, um MCE consiste em dois conjuntos de variáveis U e V, conhecidas como variáveis endógenas e exógenas, e um conjunto de funções F que atribui cada variável em V a um valor com base nos valores das outras variáveis.

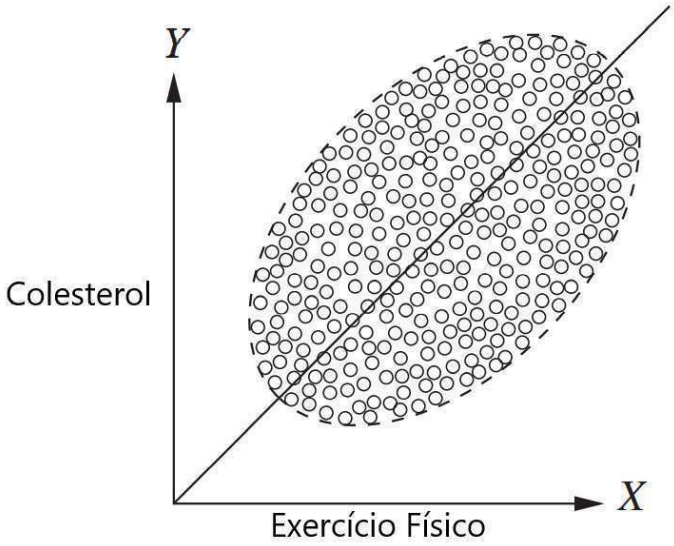


Figura 2.5: Relação entre a prática de exercícios e taxa de colesterol. Fonte: Adaptado de Pearl, Glymour e Jewell (2016).

Assim, a tripla $M(U, V, F)$ forma o MCE e todo MCE é associado a um grafo G acíclico e direcionado, em que os vértices são os conjuntos de variáveis U e V e as arestas são o conjunto de funções F.

A partir de um MCE, é possível computar relações causais entre os vértices, ou seja, é possível computar o efeito causal de um vértice A em um vértice B do grafo, desde que exista um caminho entre A e B. Esse efeito é conhecido como Efeito Causal Médio (do inglês, *Average Causal Effect* - ACE). Por exemplo, ao se considerar uma resposta binária para a pergunta "Fumar causa câncer de pulmão?", sendo 0 para Não e 1 para Sim, o ACE pode ser computado a partir da Equação 2.1 (PEARL; GLYMOUR; JEWELL, 2016).

$$ACE = E[y|do(x = 0)] - E[y|do(x = 1)], \tag{2.1}$$

em que y corresponde ao valor obtido a partir da variável de entrada x. No contexto da pergunta "Fumar causa câncer de pulmão?", x corresponde à variável *fumar* e y à resposta da pergunta. O funcional *do* corresponde a uma intervenção no modelo (PEARL; GLYMOUR; JEWELL, 2016), em que essa intervenção, no contexto algébrico, corresponde à atribuição de um valor específico à variável x. No caso da Equação 2.1 acontece a atribuição do valor *0* e *1* à variável x. No contexto de grafo, o *do* corresponde à atribuição de um valor α a um vértice V e todas as arestas que chegam ao vértice V podem ser removidas, porque os

vértices que contêm arestas direcionadas ao vértice V não influenciarão o seu valor. Em outro exemplo, a Figura 2.6 apresenta, a partir de um grafo, as relações entre temperatura, venda de sorvetes e taxa de crimes.

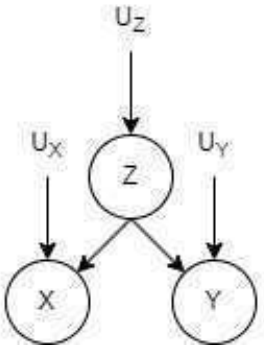


Figura 2.6: Grafo representando a relação entre temperatura (Z), venda de sorvetes (X) e taxa de crimes (Y). Fonte: Adaptado de Pearl, Glymour e Jewell (2016).

Na Figura 2.6, o vértice Z corresponde a temperatura, enquanto X e Z correspondem a vendas de sorvetes e taxa de crimes, respectivamente. Pode-se afirmar, a partir da Figura 2.6, que a temperatura causa a venda de sorvetes e a taxa de crimes. Uma intervenção para diminuir as vendas de sorvetes (fechando todas as sorveterias, por exemplo), implica em um novo grafo gerado a partir do grafo da Figura 2.6, como apresentado na Figura 2.7.

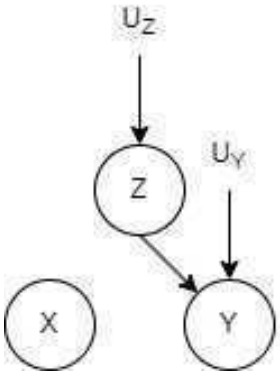


Figura 2.7: Grafo representando uma intervenção na venda de sorvete (x). Fonte: Adaptado de Pearl, Glymour e Jewell (2016).

A Figura 2.7 apresenta algo como uma cirurgia realizada no grafo apresentado na Figura 2.6, onde o vértice que corresponde à venda de sorvetes não tem nenhuma aresta chegando a ele. Neste caso, pode-se interpretar que independente da temperatura, a venda de sorvete

será baixa, que corresponde ao valor da intervenção feita nas vendas de sorvete ao fechar todas as sorveterias. Assim, pode-se concluir que a temperatura (Z), a partir do grafo da Figura 2.7 não causa a venda de sorvetes, mas continua afetando a taxa de crimes.

Capítulo 3

Trabalhos Relacionados

No presente Capítulo são apresentados os principais estudos no contexto de poda em RNA na Seção 3.1 e os principais estudos de RNA com causalidade na Seção 3.2.

3.1 Poda em RNA

Pioneiras no contexto de poda em modelo neurais, as pesquisas de LeCun, Denker e Solla (1989) e Hassibi e Stork (1992) propuseram as técnicas chamadas de *Optimal Brain Damage* (OBD) e *Optimal Brain Surgeon* (OBS), respectivamente. A partir dessas técnicas, é possível selecionar conexões de um modelo neural para serem podadas, resultando em modelos menos complexos e com boas métricas. Em ambas as técnicas é computado um valor de saliência para cada parâmetro do modelo neural, permitindo ordená-los e selecionar aqueles com baixa saliência para serem podados. A principal desvantagem dessas técnicas é a alta complexidade computacional, especialmente em modelos com muitos parâmetros. Isso ocorre em função da utilização da derivada de segunda ordem da função de perda em relação aos pesos. LeCun, Denker e Solla (1989), assim como Zhuang Liu et al. (2019), sugerem que para a obtenção de um modelo podado, faz-se necessário seguir uma sequência com três estágios, conforme a Figura 3.1.

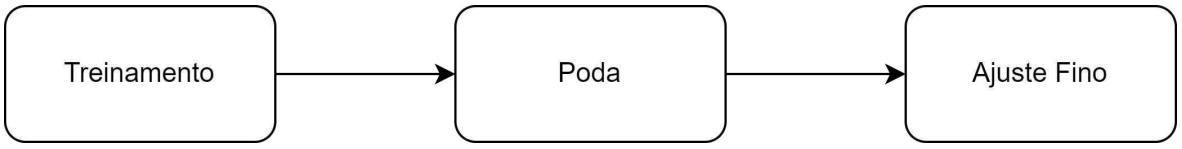


Figura 3.1: Pipeline de poda em RNA. Fonte: Adaptada de liu2018.

Na Figura 3.1, o primeiro estágio é o de treinamento do modelo, seguido da aplicação da poda e retreinamento do modelo podado considerando os pesos do modelo original treinado, conhecido como Ajuste Fino.

Han, Pool et al. (2015) propuseram uma técnica de poda que utiliza a magnitude dos pesos do modelo para selecionar as conexões que terão suas conexões com menor magnitude podadas. Diferente do treinamento convencional, onde o objetivo é ajustar os pesos das conexões para que o modelo aprenda a generalizar as saídas a partir das entradas, os autores propõem que a etapa de treinamento seja realizada para identificar quais conexões são importantes e quais podem ser podadas. Os resultados mostraram que a técnica conseguiu reduzir em 9 vezes o tamanho do modelo AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), mantendo o valor de acurácia próximo do modelo original.

Frankle e Carbin (2019) também utilizaram a magnitude dos pesos como técnica de poda no seu trabalho. Eles propuseram a técnica do bilhete de loteria, que consiste em, após a poda por magnitude, o modelo podado resultante ser retreinado a partir dos pesos iniciais do treinamento do modelo original. Frankle e Carbin (2019) presumem que existe uma subarquitetura, chamada de bilhete vencedor, contida em uma arquitetura mais complexa, que ao final do treinamento dos modelos partindo da mesma inicialização de pesos, o bilhete vencedor apresenta resultados tão bons quanto o modelo mais complexo. A Figura 3.2 apresenta as etapas até a obtenção do bilhete vencedor.

Figura 3.2: Pipeline de poda em RNA, a partir da Hipótese do bilhete vencedor, proposta por Frankle e Carbin (2019). Fonte: Autoria própria.

Na Figura 3.2, as etapas de Treinamento e Poda seguem a mesma lógica das mesmas etapas da Figura 3.1. A etapa de redefinição dos pesos, consiste em iniciar os pesos do

modelo podado a partir da inicialização do treinamento para a realização do retreinamento.

Um problema reportado por Blalock et al. (2020) refere-se à dificuldade de comparação entre técnicas de poda presentes na literatura. Isso acontece porque as pesquisas analisadas não são comparáveis entre si, pois utilizam modelos neurais e bases de dados muitas vezes diferentes. Alguns dos modelos neurais usados na literatura para avaliação de técnicas de poda são: VGG-16 (SIMONYAN; ZISSERMAN, 2014), ResNet-(18, 34, 50, 56, 110) (HE et al., 2015), entre outros. Blalock et al. (2020) fizeram um levantamento de quais os modelos neurais e bases de dados mais utilizadas por outros autores no contexto de poda, chegando à conclusão de que a combinação mais comum na literatura é a base de dados ImageNet (DENG; DONG et al., 2009) e a rede neural VGG-16. Os autores ainda afirmam que, mesmo que pesquisas usem a mesma base de dados, o mesmo modelo neural e as mesmas métricas, existirão variáveis que vão tornar as comparações entre pesquisas uma tarefa difícil, tais como: aumento de dados, pré-processamentos, variações de inicializações aleatórias, *fine-tunings*, *frameworks* de RNP, entre outros.

Por fim, Blalock et al. (2020) sugerem algumas boas práticas, como: identificar de forma clara qual a base de dados usada, métricas usadas, eficácia; utilizar pelo menos três pares de base de dados e arquitetura; informar tanto a taxa de compressão quanto a *speedup* teórica; para bases de dados com muitas classes, reportar as acurácias tanto para o top-1 quanto para o top-5; reportar as métricas tanto para o modelo original quanto para o modelo podado; plotar um gráfico apresentando o *tradeoff*, a partir da base de dados e arquitetura usadas; entre outras. Além das boas práticas apresentadas, os autores também recomendam o uso de uma biblioteca que avalia de forma padronizada o método de poda, biblioteca esta implementada por Blalock et al. (2020). Assim, novos pesquisadores na área de poda poderão facilitar a comparação de seus resultados com os de outros autores.

3.2 Causalidade

Goudet et al. (2017) propuseram as *Causal Generative Neural Networks* (CGNN), que utilizam intervenções propostas por Pearl (2009) para gerar dados que não existem no conjunto de dados de treinamento original. Os dados sintéticos, gerados a partir das CGNN, mantem as relações causais dos dados originais. Por exemplo, no contexto de geração de imagens

a partir de uma CGNN, variáveis como cor, textura e formas são usadas como entradas da CGNN, que utiliza as relações causais entre essas variáveis para gerar novos dados que se assemelham aos dados originais. Isso permite que um modelo neural, que utiliza os dados gerados a partir de uma CGNN, aprenda as relações causais entre as variáveis, o que pode ser usado para fazer inferências causais e fornecer explicações para o comportamento do modelo.

No contexto de interpretabilidade de modelos neurais, Harradon, Druce e Ruttenberg (2018) propuseram um método capaz de ajudar no entendimento sobre o comportamento dos modelos. O método consiste em realizar intervenções no código gerado por auto-codificadores para computar as relações causais entre o código produzido e a saída do modelo neural. Dessa forma, é possível obter explicações sobre o comportamento dos modelos. Para validar o método proposto, os autores aplicaram-no no contexto de classificação de imagens usando o modelo VGG-16 (SIMONYAN; ZISSERMAN, 2014). Os autores demonstraram que o método é eficaz em descobrir relações causais precisas em redes neurais profundas, tornando-as mais interpretáveis.

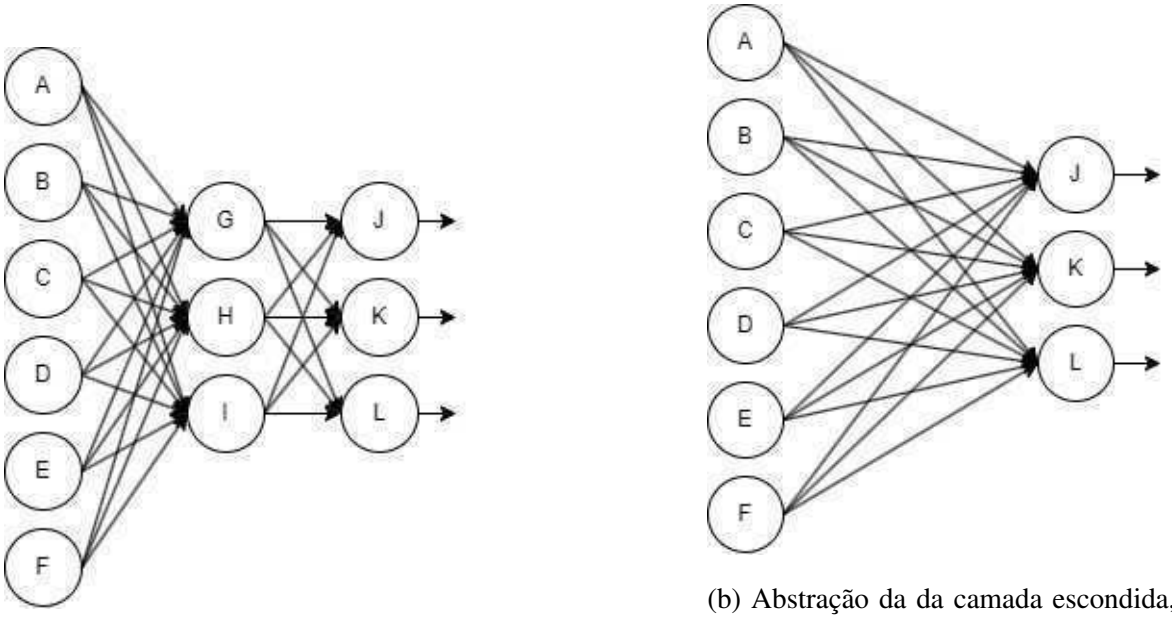
Chattopadhyay et al. (2019) utilizam causalidade para analisar o efeito das características passadas como entrada com relação à saída retornada por uma RNA. Para realizar o cálculo do efeito causal médio (ACE) das entradas de um RNA com relação à saída, Chattopadhyay et al. (2019) supõem que uma arquitetura de uma RNA pode ser vista como um MCE e a partir dele é possível computar o ACE entre neurônios. Assim, eles propuseram uma forma de computar o ACE de um neurônio da entrada do modelo com relação a um neurônio da camada de saída, objetivando obter uma interpretabilidade das características usadas na entrada com relação à saída obtida, o que, segundo os autores, faz com que os modelos neurais não sejam tratados como uma caixa preta.

A Equação 3.1, proposta por Chattopadhyay et al. (2019), tem como objetivo computar o ACE entre um neurônio da camada de entrada e um neurônio da camada de saída de uma RNA *feed forward*.

$$ACE = E[y|do(x_i = \alpha)] - E_{x_i}[E[y|do(x_i = \alpha)]], \tag{3.1}$$

em que x_i corresponde a uma entrada, α corresponde ao valor da intervenção e y a saída, a partir de $x_i = \alpha$. Segundo Chattopadhyay et al. (2019), o minuendo da subtração apresentada

no lado direito da Equação 3.1, $E[y|do(x_i = \alpha)]$, é chamado de *Inverventional Expectations* tendo sido proposto um mecanismo para realizar esse cálculo. Os autores utilizaram as seguintes bases de dados para avaliar o ACE em modelos do tipo *feed forward* entre os neurônios da camada de entrada nos neurônios da camada de saída: Iris[1] e MNIST (DENG, 2012). Como resultados obtidos, Chattopadhyay et al. (2019) verificaram quais neurônios da camada de entrada influenciam mais o resultado do modelo neural. Por exemplo, eles verificaram que apenas um neurônio é suficiente para classificar corretamente as quatro classes da base de dados Iris. Já para a base de dados MNIST, os autores realizaram o treinamento de um auto-codificador, cujo objetivo é codificar uma entrada e reconstruir a entrada a partir do código gerado, e avaliaram o ACE a partir do código gerado na reconstrução da entrada, sendo possível analisar em quais regiões da imagem cada característica do código tem um maior, e menor, ACE. Nessa pesquisa, os autores não consideram os neurônios presentes nas camadas escondidas da RNA, realizando uma abstração, conforme mostrado na Figura 3.3.



(a) RNA com 3 camadas: 1 entrada, 1 escondida e 1 saída

(b) Abstração da da camada escondida, conectando a camada de entada diretamente com a camada de saída.

Figura 3.3: Exemplo de abstração das camadas escondidas de uma RNA. Fonte Adaptada de Chattopadhyay et al. (2019).

Na Figura 3.3a, tem-se um modelo neural com 3 camadas, sendo a primeira a camada

[1]https://archive.ics.uci.edu/ml/datasets/iris, acessado em 05/05/2020 às 00h12m

de entrada, a segunda a camada escondida e a terceira a camada de saída. Já na Figura 3.3b, o modelo contém apenas duas camadas: uma de entrada e uma de saída, onde esse modelo é uma abstração do modelo da Figura 3.3a. Dessa forma, Chattopadhyay et al. (2019) conseguem computar a relação causa e enfeito entre os neurônios da camada de entrada com relação à camada de saída.

Capítulo 4

Materiais e Métodos

No presente capítulo, são apresentados todos os materiais e métodos utilizados nos experimentos, descritos na Seção 4.4. Nas Seções 4.1 e 4.2 são descritos o conjunto de dados e o modelo neural utilizados nos experimentos. Na Seção 4.3, é apresentada a abordagem de poda utilizando ACE proposta nesta pesquisa. Finalmente, na Seção 4.4, apresenta-se a metodologia adotada na condução dos experimentos.

4.1 Conjunto de Dados

De acordo com blalock2020, um dos conjuntos de dados mais utilizados para avaliação de estratégias de poda em modelos neurais profundos é o ImageNet (DENG; DONG et al., 2009). Ao todo, esse conjunto de dados contém mais de 14 milhões de imagens, divididas em mais de 21 mil classes. *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) (RUSSAKOVSKY et al., 2015) foi uma competição anual, que aconteceu entre os anos de 2010 e 2017, com dois desafios: 1) classificação de imagens, em que o objetivo era classificar as classes de objetos presentes na imagem de entrada; e 2) detecção de objetos, em que o objetivo era classificar e indicar a região da imagem de entrada que contém determinada classe. Os autores submetiam suas soluções e a que obtivesse os melhores resultados, era considerada a campeã do desafio.

O conjunto de dados experimentais utilizados na presente pesquisa foi aquele disponibilizado no ILSVRC de 2014 (RUSSAKOVSKY et al., 2015), no contexto de classificação de Imagens. ILSVRC 2014 usa um total de 1,2 milhão de imagens do banco de dados Image-

Net, divididas em 1000 classes. Entretanto, para fins de redução de tempo de processamento dos experimentos em função de limitações de hardware, para a presente pesquisa, foi considerado um subconjunto de dados do ILSVRC de 2014, com 67.317 imagens, divididas em 50 classes.

Nesta pesquisa, o conjunto de dados utilizado foi dividido em 4 subconjuntos: poda, validação, teste e treino, em que os 3 primeiros contém, cada um, 2.500 imagens e o último 59.817 imagens. O conjunto de poda é utilizado pela abordagem de poda proposta neste trabalho, com objetivo de computar o ACE entre os neurônios, conforme descrito na Subseção 4.3. O conjunto de teste é utilizado para avaliação da generalização do modelo. Já os conjuntos de validação e treino são utilizados para avaliar como está o modelo ainda na fase de treinamento e para ajustar os parâmetros do modelo, respectivamente.

Exemplos de imagens presentes no conjunto de dados utilizados nos experimentos são apresentadas na Figura 4.1. Mais detalhes acerca das classes consideradas na presente pesquisa, são apresentados no Apêndice B.

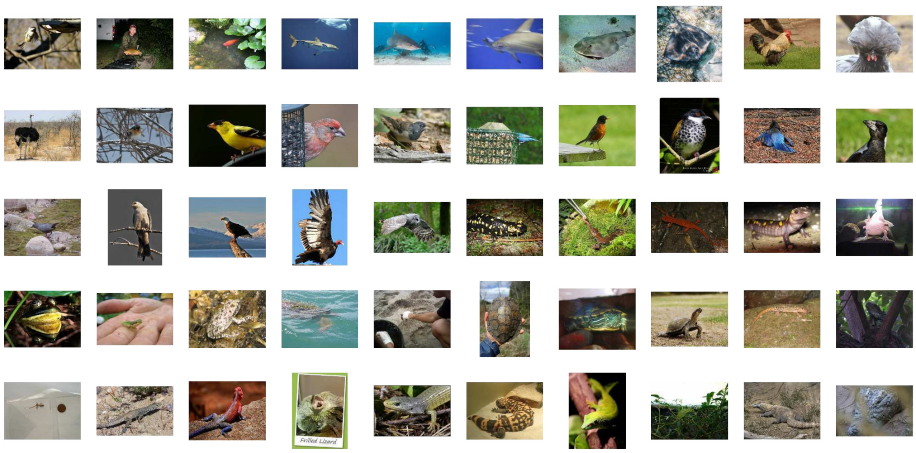


Figura 4.1: Exemplos de imagens utilizadas nos experimentos. Fonte: (RUSSAKOVSKY et al., 2015)

4.2 VGG-16

Proposta por vgg16, a VGG-16 é uma RNP que participou do ILSVRC de 2014 (RUSSAKOVSKY et al., 2015), no contexto de classificação de imagens, alcançando a segunda posição, ficando atrás apenas da GoogleNet (SZEGEDY et al., 2015), outra RNP. A Figura 4.2 apresenta a arquitetura da VGG-16.

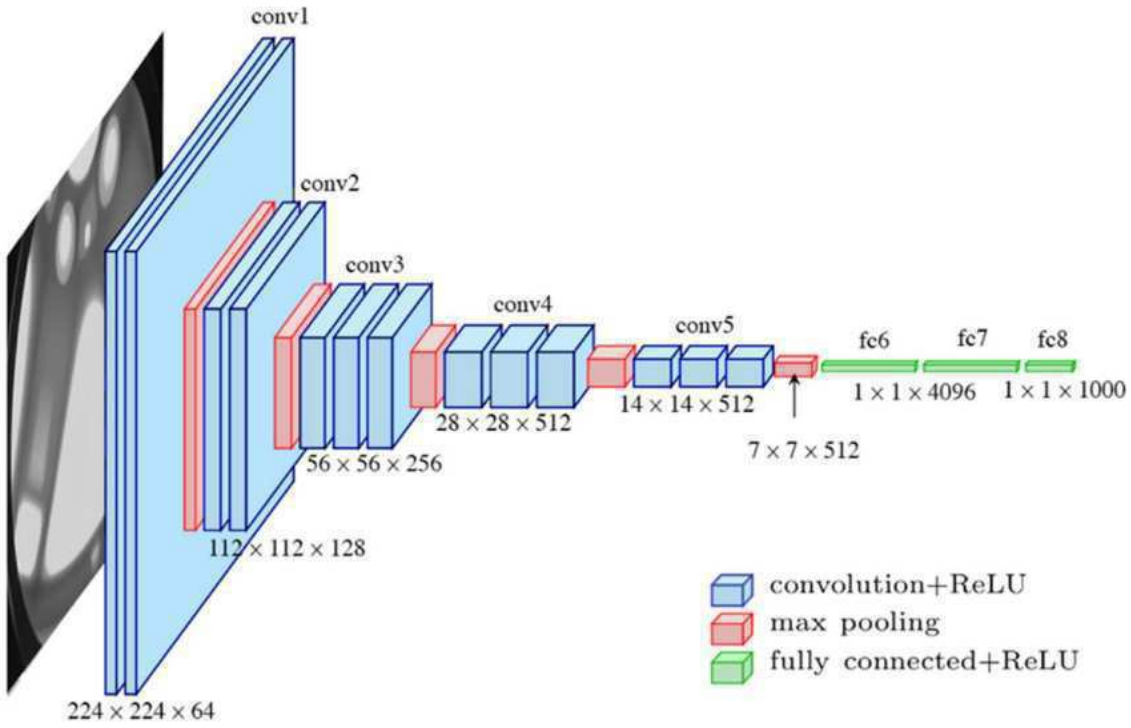


Figura 4.2: Arquitetura da VGG-16. Fonte: (FERGUSON et al., 2017)

A partir da Figura 4.2, percebe-se que ela contém: 13 camadas convolucionais, 5 de *polling* e 3 camadas completamente conectadas. A imagem de entrada da VGG-16 tem resolução de 224 $\times$ 224 pixeis e a saída contém 1000 neurônios, onde cada neurônio representa uma classe. Ao todo, a VGG-16 contém mais de 130 milhões de parâmetros e a maioria destes parâmetros (mais de 123 milhões) pertence às camadas completamente conectadas. Para a presente pesquisa, foi considerada uma arquitetura com 50 neurônios na camada de saída, para atender ao conjunto de dados descrito na Seção 4.1, que corresponde a um subconjunto da ImageNet, com 50 classes, resultando em um modelo com 119.750.706 parâmetros.

4.3 Técnica de poda a partir do ACE entre neurônios

A técnica de poda proposta na presente pesquisa é aplicada apenas às camadas escondidas e completamente conectadas da VGG-16. Essa poda é considerada uma poda estruturada, já que neurônios e todas as suas conexões associadas são removidos após a poda, conforme descrito na Subseção 2.2. A Figura 4.3 apresenta os passos do método de poda proposto, desde o treinamento até o modelo podado.

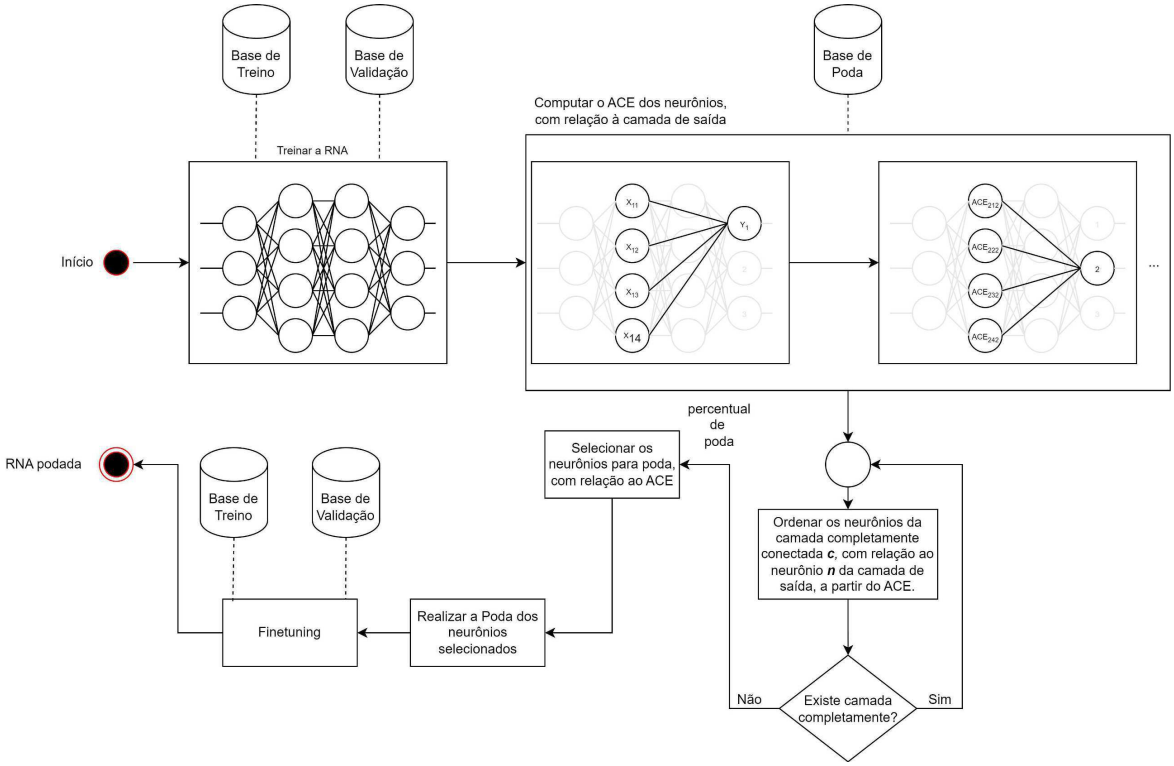


Figura 4.3: Fluxo considerado desde o treinamento de uma RNA até o modelo podado. Fonte: Autoria própria.

Na Figura 4.3 é possível visualizar todas as etapas necessárias para realizar a poda a partir do ACE entre os neurônios. As etapas são descritas a seguir:

Treinamento do modelo neural

O modelo VGG-16, apresentado na Seção 4.2, é treinado a partir do conjunto de dados de treinamento e a perda monitorada no conjunto de validação, com vistas e reduzir o risco de *overfitting*. Nesta etapa, os pesos do modelo são iniciados de forma aleatória e ajustados,

ao longo do treinamento, até que o modelo possa generalizar para outros dados diferentes do treinamento. Ao final de cada época (após todos os dados do conjunto de treinamento serem passados para o modelo uma vez), são computadas as seguintes métricas: acurácia do conjunto de validação e de treinamento, e o erro obtido.

Cálculo do ACE entre os neurônios

O cálculo do ACE é a principal etapa para realização da abordagem de poda proposta e acontece após o treinamento do modelo VGG-16. Em primeiro lugar, é necessário passar para o modelo treinado um conjunto de dados de poda para calcular o ACE entre os neurônios das camadas completamente conectadas, com relação à camada de saída.

Para computar o ACE entre neurônios, é preciso discretizar os neurônios das camadas completamente conectadas em ativado e não ativado. Sabe-se que a função de ativação usada nas camadas completamente conectadas do modelo VGG-16 é RELU, onde o valor do neurônio corresponde ao máximo entre o valor de entrada e zero, ou seja, o valor do neurônio é um número real positivo. O primeiro passo para discretizar os valores dos neurônios é computar a média e desvio padrão dos neurônios das camadas completamente conectadas, a partir do conjunto de poda, como apresentado no Algoritmo 1. Onde, para cada imagem do conjunto de dados de poda é realizado um *forward* - processo de propagar os dados de entrada através do modelo VGG-16, a partir da camada de entrada, para produzir uma classificação - e todas as ativações dos neurônios das camadas completamente conectadas são computadas. Em que, um neurônio é considerado ativado quando o seu respectivo valor está acima da média em pelo menos dois desvios padrão, conforme a Equação 4.1.

Algoritmo 1 Cálculo da média e desvio padrão dos neurônios. **Entrada**: $conjunto\_de\_poda, modelo$

$valores\_relu_por_camada \leftarrow \{\}$

para $imagem$ **em** $conjunto\ de\ poda$ **faça**

$resultado \leftarrow modelo.forward(imagem)$

para fc **em** $resultado.camadas\_completamente\_conectadas()$ **faça**

para $neuronio$ **em** $camada\_completamente\_conectada$ **faça**

$valores\_relu\_por\_camada[fc].append(neuronio)$

fim para

fim para

fim para

$media \leftarrow \{\}$

$desvio\_padrao \leftarrow \{\}$

para fc **em** $modelo.camadas\_completamente\_conectadas()$ **faça**

$media[fc] = computar\_media(valores\_relu\_por\_camada[fc])$

$desvio\_padrao[fc] = computar\_desvio\_padrao(valores\_relu\_por\_camada[fc])$

fim para

retorno $media, desvio\_padrao$

No Algoritmo 1, são apresentados os passos necessários para o cálculo da média e desvio padrão dos neurônios das camadas completamente conectadas.

$$a = \begin{cases} 1 & se \quad (x - \bar{x})/std) \geq 2 \\ 0 & c.c \end{cases} \tag{4.1}$$

Na Equação 4.1, a corresponde à ativação do neurônio, onde 1 significa que o neurônio foi ativado e 0 que o neurônio não foi ativado. Já x, $\bar{x}$ e std correspondem ao valor do neurônio, o valor da média e o valor do desvio padrão, respectivamente. Para a escolha do limiar 2, foram realizados alguns experimentos variando esse limiar, onde os resultados

podem ser encontrados no Apêndice C.

Finalmente, a partir das ativações dos neurônios, é possível calcular o ACE de um neurônio de uma camada escondida com relação a um neurônio da camada de saída, conforme a Equação 4.2.

$$ACE_{l_{i_{Y=y_j}}} = P(Y = y_j|do(X_{l_i} = 1)) - P(Y = y_j|do(X_{l_i} = 0)). \tag{4.2}$$

Na Equação 4.2, l corresponde a uma camada completamente conectada, i indica um neurônio da camada l, y_j se refere a um neurônio da camada de saída, ou seja, a uma classe. Portanto, $ACE_{l_{i_{y=Y_j}}}$ corresponde ao ACE do neurônio i da camada completamente conectada l com relação à classe Y_j. Por outro lado, $x_{l_i} = 1$ significa que o neurônio i da camada l está ativado, enquanto $x_{l_i} = 0$ o neurônio não está ativado. Logo, $P(Y = y_j|do(X_{l_i} = 1))$ corresponde a probabilidade de Y ser da classe Y_j dado a intervenção de $X_{l_i} = 1$, enquanto $P(Y = y_j|do(X_{l_i} = 0))$ corresponde a probabilidade de Y também ser da classe Y_j dado a intervenção de $X_{l_i} = 0$.

O cálculo do minuendo e subtraendo da diferença presente no lado direito da Equação 4.2 é conseguido a partir das Equações 4.3 e 4.4, respectivamente.

$$P(Y = y_j|do(X_{l_i} = 1)) = \sum_z P(Y = y_j|X_{l_i} = 1, PA = z)P(PA = z) \tag{4.3}$$

$$P(Y = y_j|do(X_{l_i} = 0)) = \sum_z P(Y = y_j|X_{l_i} = 0, PA = z)P(PA = z) \tag{4.4}$$

Nas Equações 4.3 e 4.4, PA corresponde aos neurônios pais de X, ou seja, todos os neurônios da camada anterior.

Por exemplo, para computar o ACE do neurônio X_{21}, com relação à saída Y_1, do modelo apresentado na Figura 4.4, o PA corresponde aos neurônios pais de X_{21}, ou seja, Z_1, Z_2 e Z_3.

Dessa forma, a equação para computar o ACE do neurônio X_{21}, é a seguinte:

$ACE_{X_{21_{Y=y_1}}} = P(Y = y_1|do(X_{21} = 1)) - P(Y = y_1|do(X_{21} = 0))$

em que:

$P(Y = y_1|do(X_{21} = 1)) = \sum_z P(Y = y_1|X_{21} = 1, PA = z)P(PA = z)$

$P(Y = y_1|do(X_{21} = 0)) = \sum_z P(Y = y_1|X_{21} = 0, PA = z)P(PA = z)$

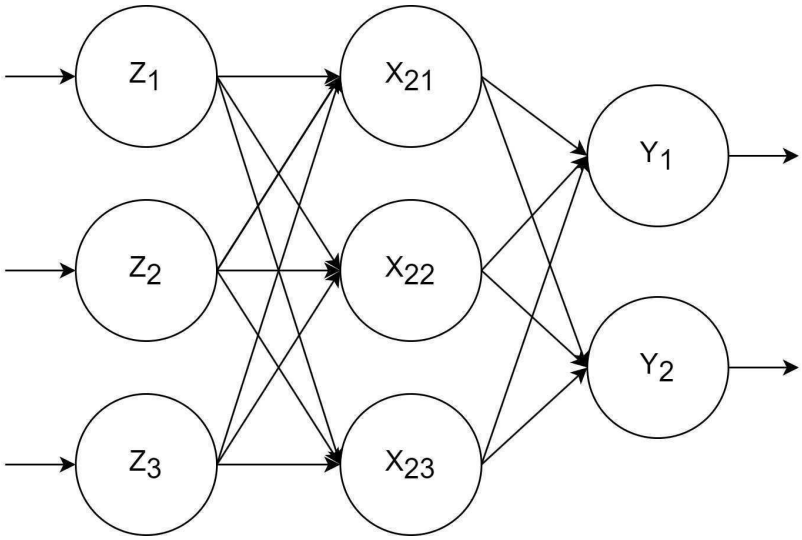


Figura 4.4: Modelo neural com 3 camadas, sendo uma de entrada, uma escondida e outra de saída. Fonte: Autoria própria.

como $PA = (Z_1, Z_2, Z_3)$, tem-se:

$$
\begin{aligned}
&P(Y = y_1|do(X_{21} = 1)) = \\
&P(Y = y_1|X_{21} = 1, Z_1 = 0, Z_2 = 0, Z_3 = 0)P(Z_1 = 0, Z_2 = 0, Z_3 = 0)+ \\
&P(Y = y_1|X_{21} = 1, Z_1 = 0, Z_2 = 0, Z_3 = 1)P(Z_1 = 0, Z_2 = 0, Z_3 = 1)+ \\
&P(Y = y_1|X_{21} = 1, Z_1 = 0, Z_2 = 1, Z_3 = 0)P(Z_1 = 0, Z_2 = 1, Z_3 = 0)+ \\
&P(Y = y_1|X_{21} = 1, Z_1 = 0, Z_2 = 1, Z_3 = 1)P(Z_1 = 0, Z_2 = 1, Z_3 = 1)+ \\
&P(Y = y_1|X_{21} = 1, Z_1 = 1, Z_2 = 0, Z_3 = 0)P(Z_1 = 1, Z_2 = 0, Z_3 = 0)+ \\
&P(Y = y_1|X_{21} = 1, Z_1 = 1, Z_2 = 0, Z_3 = 1)P(Z_1 = 1, Z_2 = 0, Z_3 = 1)+ \\
&P(Y = y_1|X_{21} = 1, Z_1 = 1, Z_2 = 1, Z_3 = 0)P(Z_1 = 1, Z_2 = 1, Z_3 = 0)+ \\
&P(Y = y_1|X_{21} = 1, Z_1 = 1, Z_2 = 1, Z_3 = 1)P(Z_1 = 1, Z_2 = 1, Z_3 = 1)
\end{aligned}
$$

O mesmo raciocínio é considerado para computar $P(Y = y_1|do(X_{21} = 0))$. A partir do exemplo acima, é possível perceber que é preciso considerar todas as combinações possíveis de PA. Assim, para situações onde PA é um conjunto com muitos elementos, esse cálculo se torna mais demorado e o resultado pode ser um número bem pequeno.

No contexto do modelo VGG-16, a quantidade de neurônios que fazem parte de PA é, no mínimo, 4096. Ou seja, é esperado que os valores do ACE dos neurônios sejam pequenos. Uma forma de otimizar o cálculo do ACE, é considerar apenas as combinações de PA que apareceram durante o cálculo das ativações dos neurônios. Por exemplo, supondo que $(Z_1 =$

$1, Z_2 = 1, Z_3 = 1)$ seja uma combinação que nunca aconteceu, a parcela do somatório que contém essa combinação será igual a $zero$, assim o cálculo dessa parcela poderá ser evitado. Um exemplo, demonstrado por Pearl, Glymour e Jewell (2016) para o cálculo do ACE, é apresentado no Apêndice A.

Ordenação dos neurônios

Finalmente, após o cálculo dos ACE entre os neurônios das camadas completamente conectadas com relação à saída, descrito na etapa anterior, para cada camada completamente conectada, os seus respectivos neurônios são ordenados de forma crescente a partir do valor do ACE, ou seja, a ordenação é feita do neurônio com menor impacto causal até o neurônio com maior impacto causal.

Seleção dos neurônios a serem podados

Para cada camada completamente conectada, um percentual dos neurônios é escolhido para sofrer poda, sendo que os neurônios escolhidos são os que têm menores valores de ACE, ou seja, para uma poda de 10%, por exemplo, os 10% de neurônios de cada camada completamente conectada com menores ACE serão selecionados.

Aplicação da poda

Nesta etapa, os neurônios selecionados na etapa anterior são removidos do modelo neural, assim como todas as suas respectivas conexões. Ou seja, é aplicada uma poda estruturada, como descrita na Seção 2.2. O modelo neural resultante desta etapa é um modelo menos complexo do que o modelo original. Por exemplo, ao considerar o modelo da Figura 4.4 como modelo original, após uma poda estruturada, onde o neurônio X_{22} é podado, o modelo resultante seria conforme o apresentado na Figura 4.5. Percebe-se que a remoção de um neurônio, nesse exemplo, implica a remoção de 5 parâmetros, correspondendo a cerca de 30% dos parâmetros do modelo. Considerando o modelo VGG-16, por exemplo, onde as camadas completamente conectadas possuem milhares de neurônios, o impacto tende a ser significativo.

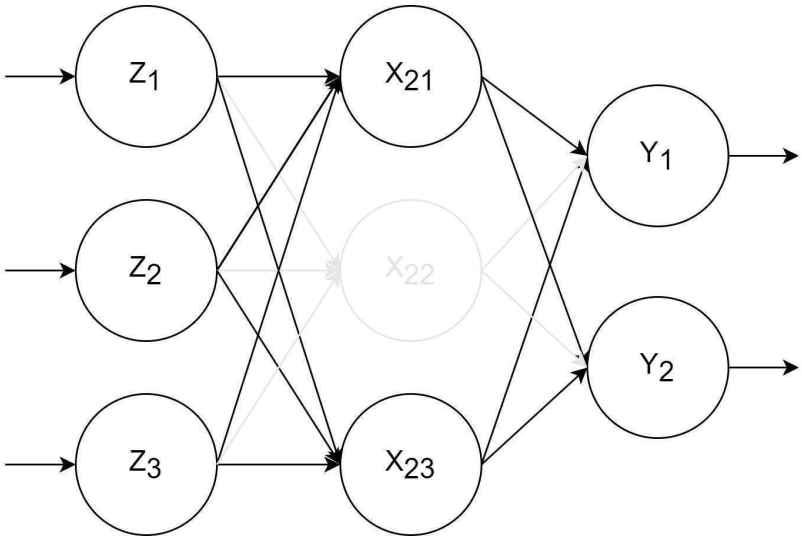


Figura 4.5: Exemplo de poda. Fonte: Autoria própria

Ajuste Fino

Finalmente, a última etapa consiste em um retreinamento do modelo neural, sobre o conjunto de treinamento e com controle de generalização pelo conjunto de validação, tendo como pesos iniciais aqueles ajustados na etapa de treinamento original. A diferença desta etapa para a etapa de treinamento, é que na etapa de treinamento os pesos do modelo são iniciados de forma aleatória, enquanto nesta etapa a inicialização dos pesos acontece a partir do treinamento realizado inicialmente.

4.4 Plano Experimental

A validação experimental da abordagem de poda proposta na presente pesquisa foi estruturada nas seguintes etapas: treinamento, poda e avaliação dos modelos neurais produzidos. Para fins de comparação, foram também experimentadas duas técnicas de poda: por Magnitude dos pesos e seleção aleatória de pesos, conforme descrito na Subseção 3.1. Ao final da etapa de treinamento, tem-se 4 modelos resultantes: o modelo original sem poda, o modelo resultante da técnica de poda proposta, e os modelos resultantes da poda por Magnitude e aleatória. Os seguintes percentuais de poda foram considerados na avaliação experimental: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% e 95%.

Na etapa de avaliação de cada modelo gerado, utilizou-se a técnica de *bootstrap* com

$n = 30$ repetições, sendo uma abordagem para ajustar a variação de uma estimativa baseada em uma amostra, ao invés de uma população inteira (EFRON, 1979), a partir do conjunto de teste para computar as respectivas acurácias e intervalos de confiança, além do tempo médio de predição. Também foram avaliados a quantidade de parâmetros das arquiteturas após as podas e o espaço em disco ocupado por cada modelo, após salvá-los em disco a partir do método de salvamento do próprio Pytorch[1], onde são salvos a arquitetura do modelo e a matriz de pesos, considerando todas as camadas do modelo.

Os experimentos foram conduzidos em um computador com as configurações a seguir: Ubuntu 16.04.6 LTS Linux Kernel 4.15.0-107-generic, processador Intel ©Core™i7-8700K CPU @ 3.70GHz, memória RAM 2×16GiB @ 2666 MHz e GPU GeForce 2080 RTX Ti com 11GB. Para implementação dos fluxos de treinamento e teste do modelo VGG-16, utilizou-se a linguagem de programação Python v3.7.9 e o *framework* PyTorch[2]. Para a poda por Magnitude dos pesos, foi utilizado o módulo *prunne*[3] do Pytorch. Todo o código produzido está disponível em um repositório público[4].

[1]https://pytorch.org/tutorials/beginner/saving_loading_models.html, acessado em Maio de 2023

[2]https://pytorch.org/, acessado em Abril de 2023

[3]https://pytorch.org/docs/stable/generated/torch.nn.utils.prune.ln_structured.html, acessado em Abril 2023

[4]https://github.com/carlosInteraminense/ACE4pruning, acessado em Abril de 2023

Capítulo 5

Resultados e Discussões

No presente capítulo são apresentados e discutidos os resultados conforme o plano experimental apresentado na Subseção 4.4. Nas Seções 5.1, 5.2, 5.3 e 5.4 são apresentadas avaliações comparativas das acurácias, tempos de resposta, quantidade de parâmetros e espaço em disco ocupados, respectivamente, obtidos a partir dos modelos podados via diferentes técnicas, além do próprio modelo original.

5.1 Avaliação Comparativa das Acurácias

As Figuras 5.1, 5.2 e 5.3 apresentam as acurácias obtidas pelas técnicas de poda por ACE, magnitude dos pesos e seleção aleatória de neurônios, respectivamente, ao serem considerados diferentes percentuais de poda. Para cada técnica e percentual de poda, foram computados os intervalos de confiança das acurácias obtidas.

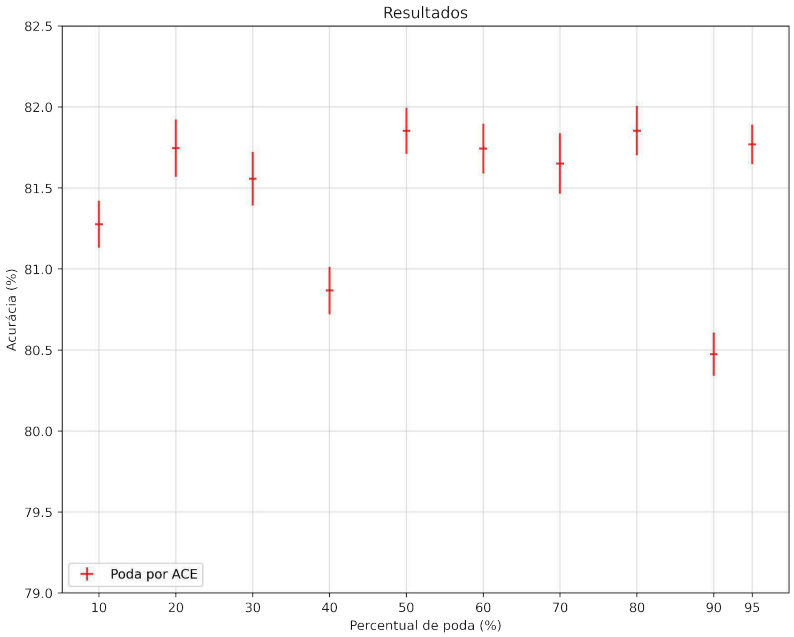


Figura 5.1: Intervalos de confiança das acurácias obtidas a partir da poda por ACE em função dos diferentes percentuais de poda investigados (10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% e 95%).

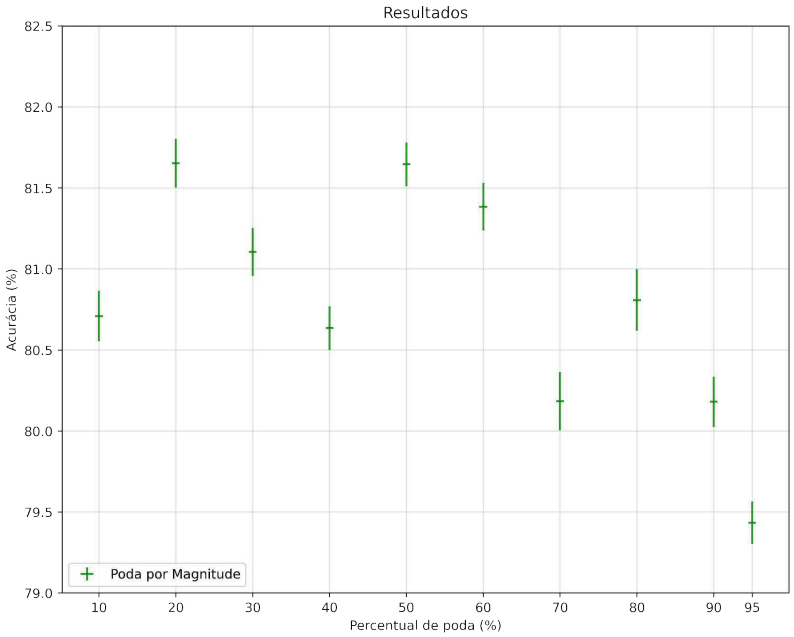


Figura 5.2: Intervalos de confiança das acurácias obtidas a partir da poda por magnitude dos pesos em função dos diferentes percentuais de poda investigados (10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% e 95%).

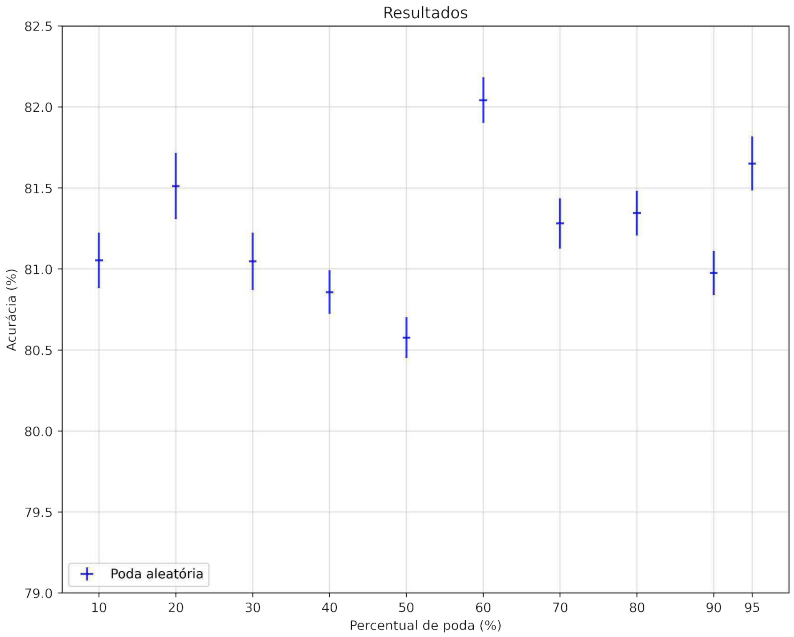


Figura 5.3: Intervalos de confiança das acurácias obtidas a partir da poda por seleção aleatória de neurônios em função dos diferentes percentuais de poda investigados (10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% e 95%).

A Figura 5.4, apresenta uma sobreposição dos intervalos de confiança das acurácias, considerando todas as técnicas de poda analisadas, que foram apresentadas nas Figuras 5.1, 5.2 e 5.3. Os casos em que há uma intersecção entre os intervalos de confiança de uma ou mais técnicas de poda, para o mesmo percentual de poda, não é possível afirmar que há diferença estatística entre os resultados. Para os casos em que não há intersecção, quanto maiores os valores da acurácia melhor será a técnica.

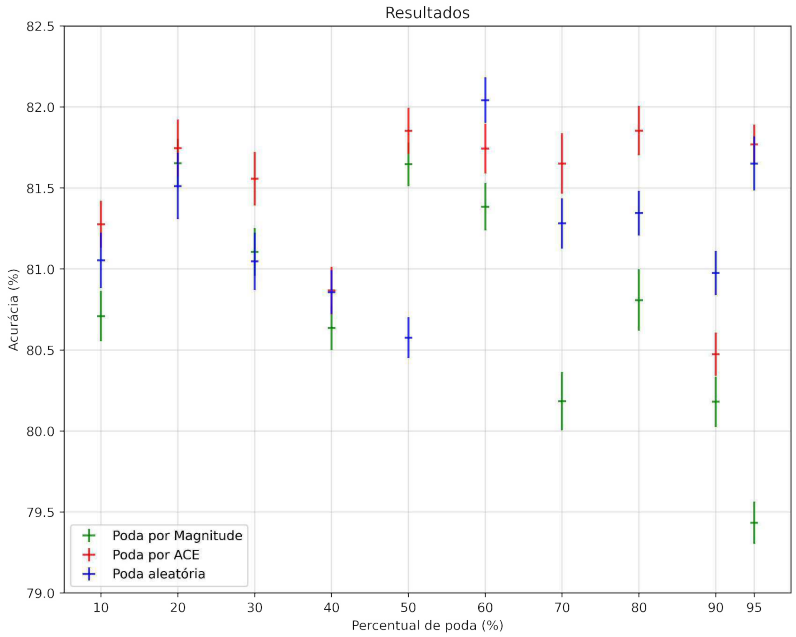


Figura 5.4: Resultado aglutinando os intervalos de confiança de acurácia das 3 técnicas de poda avaliadas.

A partir da Figura 5.4, é possível perceber que não existe uma técnica de poda dominante para todos os percentuais de poda aplicados. Contudo, o método proposto na presente pesquisa, poda por ACE, possui melhores resultados para 30%, 70% e 80% de poda e está empatado, no primeiro lugar, com outras técnicas em 10%, 20%, 40%, 50%, e 95% de poda. Enquanto a técnica que escolhe os neurônios aleatoriamente, vence apenas ao se considerar um percentual de poda de 60% e 90%. Dessa forma, a poda utilizando o ACE como técnica apresenta os melhores resultados em oito percentuais de poda, de um total de dez analisados. Também é possível observar, a partir da Figura 5.4, que a técnica de poda por magnitude dos pesos, apresenta os piores resultados de acurácia, assumindo o último lugar em quase todos os percentuais de poda analisados e estando em primeiro lugar apenas quando o percentual de poda aplicado foi 20% e 50%, onde compartilha o primeiro lugar com a técnica de poda usando ACE e a seleção aleatória de neurônios, para 20% de poda, e o ACE para 50% de poda.

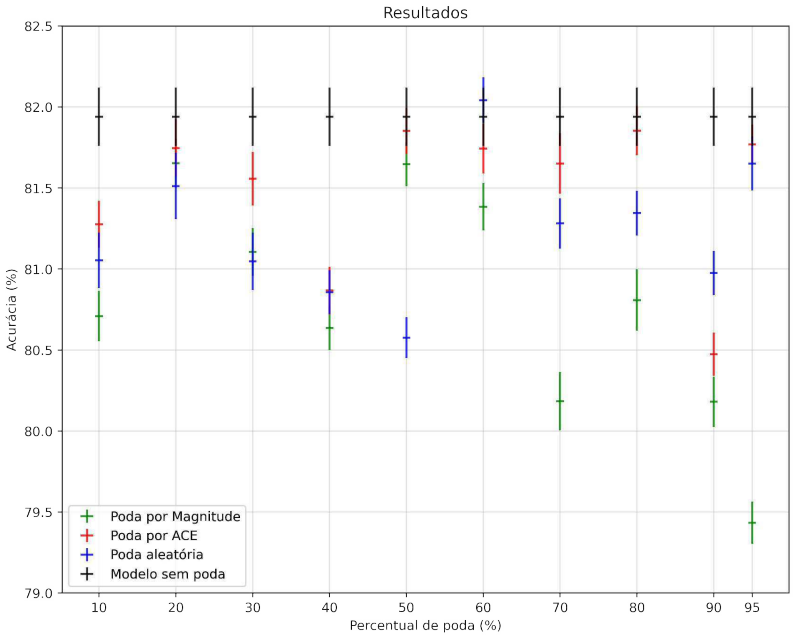


Figura 5.5: Resultado aglutinando os intervalos de confiança de acurácia das 3 técnicas de poda avaliadas e o modelo sem poda.

Para fins de comparação com as técnicas de poda analisadas, na Figura 5.5, os resultados obtidos com o modelo sem poda são repetidos ao longo dos percentuais de podas analisados. Desta forma, é possível perceber que o modelo sem poda apresenta os melhores resultados de acurácia. Contudo, para 20%, 50%, 60%, 70%, 80% e 95% de poda, a técnica de poda por ACE apresenta intersecção dos intervalos de confiança com o modelo sem poda. Assim, é possível concluir que as acurácias obtidas com o modelo sem poda são estatisticamente iguais aos obtidos pelos modelos podados a partir do ACE entre os neurônios, na maioria dos percentuais de poda analisados.

5.2 Avaliação Comparativa dos Tempos de Resposta

As Figuras 5.6, 5.7 e 5.8 apresentam os tempos para inferência sobre uma única imagem obtidos para cada nível considerando as técnicas de poda por ACE, magnitude dos pesos e seleção aleatória de neurônios, respectivamente. Já a Figura 5.9, apresenta o resultado de tempo para o modelo sem poda, ou seja, o modelo base (VGG-16). Para cada técnica e

percentual de poda, os tempos de respostas foram representados na forma de boxplots, para fins de análise.

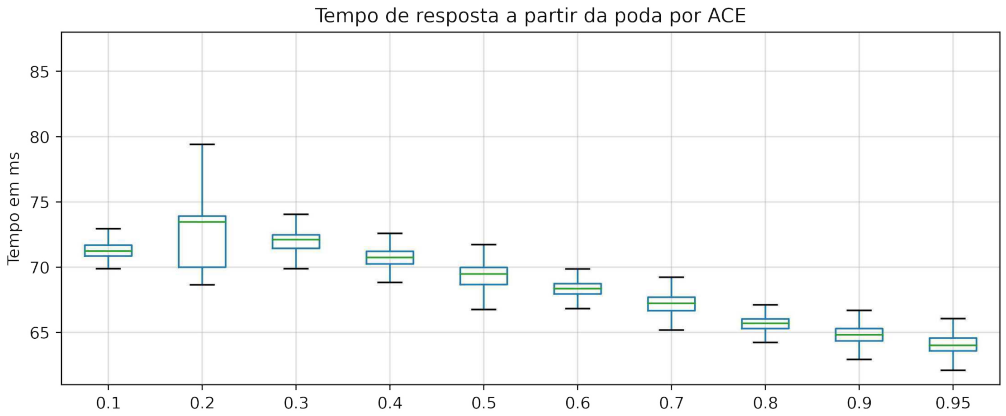


Figura 5.6: Tempo de resposta para inferência sobre uma única imagem dos modelos, a partir da poda por ACE

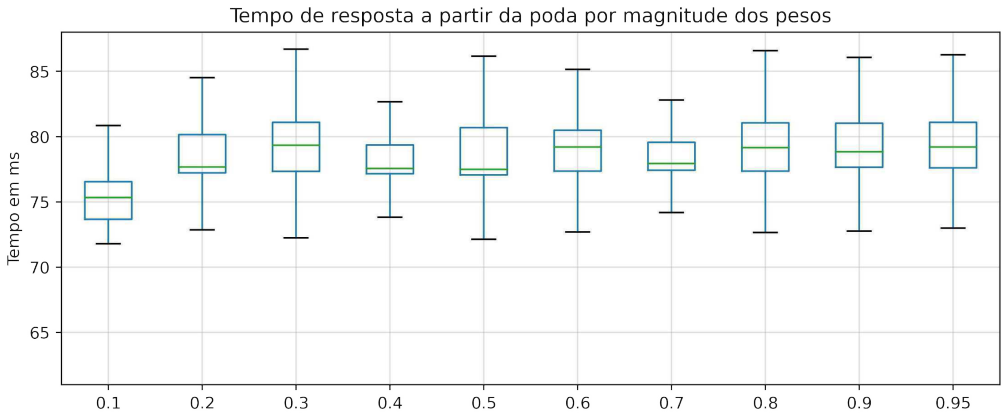


Figura 5.7: Tempo de resposta para inferência sobre uma única imagem dos modelos, a partir da poda por magnitude dos pesos

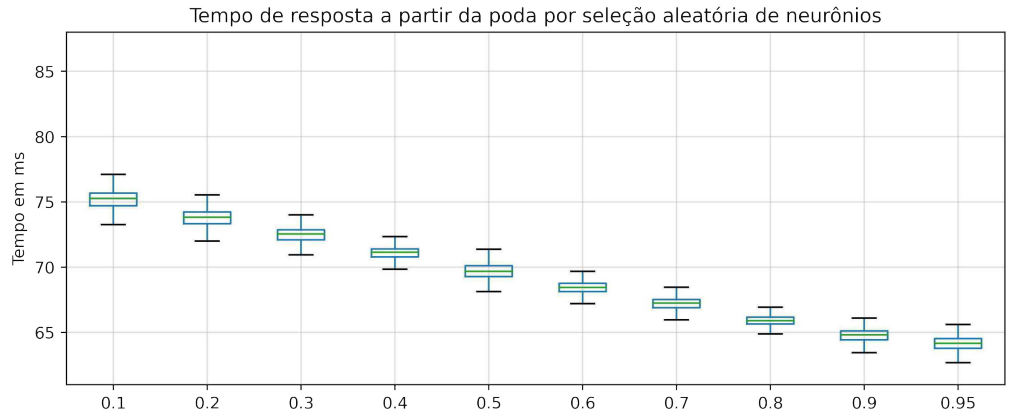


Figura 5.8: Tempo de resposta para inferência sobre uma única imagem dos modelos, a partir da poda por seleção aleatório de neurônios

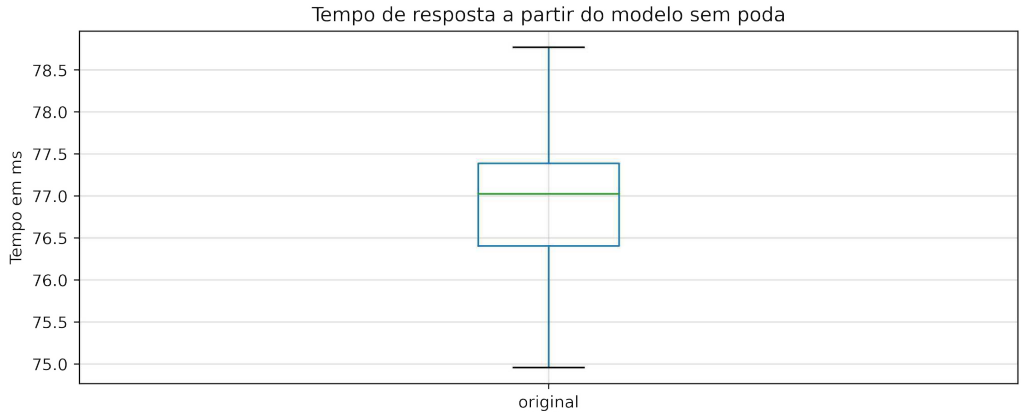


Figura 5.9: Tempo de resposta para inferência sobre uma única imagem do modelo sem poda

A partir das Figuras 5.6 e 5.8, que correspondem a técnica de poda por ACE e por seleção aleatória de neurônios, respectivamente, é possível perceber que o tempo diminui na medida que o percentual de poda aumenta. Já, ao considerar a Figura 5.7, que corresponde a técnica de poda a partir da magnitude dos pesos, percebe-se que o tempo não varia muito, independente do percentual de poda aplicado. Finalmente, na Figura 5.9, pode-se perceber que o valor do tempo de resposta está próximo dos $77ms$.

A Figura 5.10, apresenta uma sumarização dos tempos de respostas de todas as técnicas de poda analisadas na presente pesquisa, bem como o modelo sem poda, considerando a mediana dos boxplots das Figuras 5.6, 5.8 e 5.7.

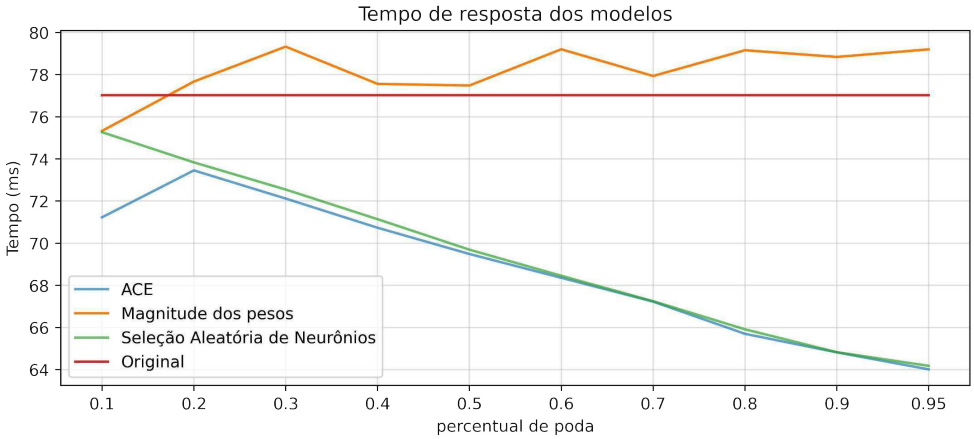


Figura 5.10: Mediana dos tempos de resposta

A partir da Figura 5.10, é possível perceber que o tempo de resposta dos modelos após a poda por magnitude é superior aos demais modelos. Os modelos que foram podados com ACE ou por seleção aleatória de neurônios apresentaram tempos de resposta inferiores ao do modelo sem poda, e ambas as técnicas resultam em modelos com tempo de resposta equivalentes. Apesar da poda por magnitude dos pesos resultar em um modelo neural menos complexo, o tempo de resposta, após diferentes percentuais de poda, é equivalente ao modelo original. Isso acontece devido ao retorno do modelo podado a partir do módulo *prunne*[1] do Pytorch conter a mesma quantidade de parâmetros do modelo original, dando indícios de ser uma poda não estruturada, ou seja, resultando em um modelo esparso, em que zeros são atribuídos aos pesos das conexões podadas.

5.3 Quantidade de parâmetros dos modelos

A Figura 5.11 apresenta a quantidade de parâmetros dos modelos após a aplicação das diferentes técnicas de poda, bem como a quantidade de parâmetros do modelo sem poda.

[1]https://pytorch.org/docs/stable/generated/torch.nn.utils.prune.ln_structured.html, acessado em Abril de 2023

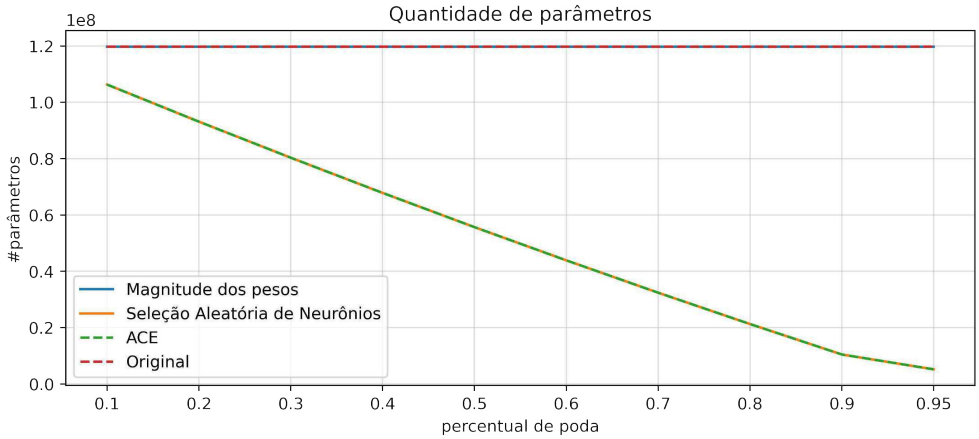


Figura 5.11: Quantidade de parâmetros dos modelos após as podas

A partir da Figura 5.11, é possível perceber que a quantidade de parâmetros do modelo sem poda e dos resultados das podas por Magnitude são os mesmos. Já as quantidades de parâmetros dos modelos podados a partir do ACE e Aleatório também são iguais, mas diminuem conforme o percentual de poda aumenta.

5.4 Espaço em Disco Ocupado após as podas

A Figura 5.12 apresenta o espaço em disco ocupado pelos modelos após as podas e o modelo sem poda, em que os modelos completos foram salvos em disco a partir da função *save* do Pytorch[2].

[2]https://pytorch.org/tutorials/beginner/saving_loading_models.html, acessado em Abril de 2023

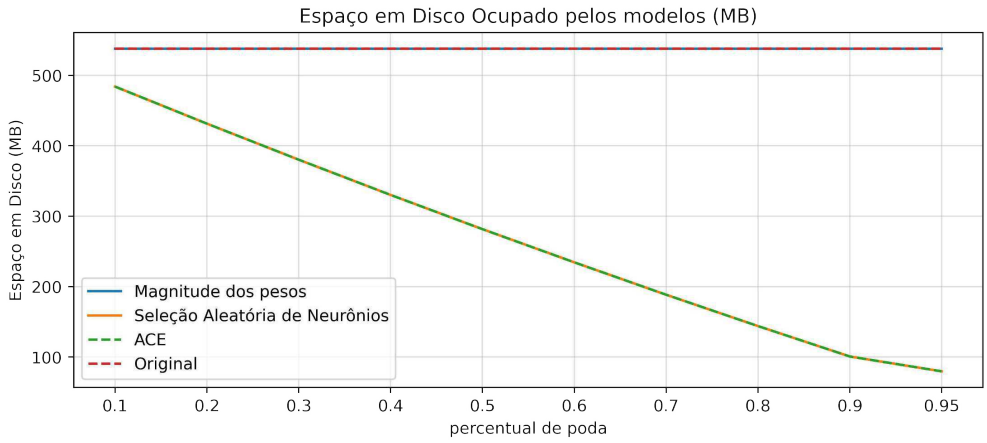


Figura 5.12: Espaço em disco ocupado pelos modelos após as podas

A partir da Figura 5.12, é possível perceber que o espaço em disco ocupado pelo modelo sem poda e dos resultados das podas por Magnitude são os mesmos. O espaço em disco ocupado pelos modelos podados a partir da poda por ACE e por seleção aleatória de neurônios também são iguais, mas diminuem conforme o percentual de poda aumenta.

5.5 Considerações Finais

A Tabela 5.1 apresenta uma compilação dos resultados obtidos a partir das técnicas de podas analisadas, comparando seus respectivos resultados com os resultados obtidos a partir do modelo original. Para cada métrica, foi computado o ganho ao realizar podas no modelo original. Assim, quanto maior o valor das células, maior foi o ganho obtido. Na Tabela 5.1, as células que contém o símbolo ↑ indicam a célula que obteve o melhor resultado entre as técnicas de poda analisadas. O valor 0 (zero), indica que a técnica apresentou resultados iguais aos obtidos pelo modelo original. Já os valores negativos e positivos, indicam que a técnica apresentou resultados piores e melhores do que o modelo original, respectivamente.

Tabela 5.1: Comparação dos resultados obtidos com os resultados do modelo original

Métrica	Percentual de poda	Poda por ACE	Poda por magnitude dos pesos	Poda por seleção aleatória de neurônios
Acurácia (%)	**10**	-0,34↑	-0,90	-0,53
	20	0,00 ↑	0,00 ↑	-0,04
	30	-0,04 ↑	-0,51	-0,54
	40	-0,75 ↑	-0,99	-0,77
	50	0,00 ↑	0,00 ↑	-1,06
	60	0,00 ↑	-0,23	0,00 ↑
	70	0,00 ↑	-1,40	-0,32
	80	0,00 ↑	-0,76	-0,28
	90	-1,15	-1,42	-0,65 ↑
	95	0,00 ↑	-2,20	0,00 ↑
Tempo de inferência (ms)	**10**	5,80 ↑	1,70	1,76
	20	3,57 ↑	-0,65	3,19
	30	4,91 ↑	-2,30	4,48
	40	6,29 ↑	-0,53	5,89
	50	7,54 ↑	-0,46	7,33
	60	8,67 ↑	-2,18	8,57
	70	9,80 ↑	-0,91	9,77
	80	11,32 ↑	-2,13	11,11
	90	12,20 ↑	-1,81	12,19
	95	13,01 ↑	-2,17	12,84
Espaço em disco ocupado (MB)	**10**	53,90 ↑	0	53,90 ↑
	20	106,50 ↑	0	106,50 ↑
	30	157,70 ↑	0	157,70 ↑
	40	207,70 ↑	0	207,70 ↑
	50	256,30 ↑	0	256,30 ↑
	60	303,40 ↑	0	303,40 ↑
	70	349,40 ↑	0	349,40 ↑
	80	393,90 ↑	0	393,90 ↑
	90	437,10 ↑	0	437,10 ↑
	95	458,20 ↑	0	458,20 ↑

A partir dos resultados apresentados na Tabela 5.1, pode-se perceber que a técnica de

poda apresenta os melhores resultados, quando comparados com os resultados obtidos a partir das outras técnicas de poda. No contexto da acurácia, para os percentuais de poda de 20%, 50%, 60%, 70%, 80%e 95%, a técnica de poda por ACE apresentou resultados iguais aos obtidos pelo modelo original, já para os demais percentuais de poda houve perdas de acurácia, onde a maior perda aconteceu no percentual de poda igual a 90%, com uma perda de 1,15%. Com relação ao tempo de inferência, a técnica de poda também apresentou os melhores resultados, superando os resultados obtidos no modelo original, em todos os percentuais de poda. Finalmente, no contexto de espaço em disco, os resultados obtidos tanto a partir da poda por ACE quando a partir da poda por seleção aleatória dos neurônios, foram iguais. Percebe-se, também, que com relação ao tempo de inferência e espaço em disco ocupado, quanto mais agressiva for a poda, maiores serão os ganhos do modelo podado.

Capítulo 6

Considerações Finais

No presente capítulo, são apresentadas as principais conclusões e contribuições desta pesquisa, bem como possíveis trabalhos futuros que podem ser realizados a partir da pesquisa realizada nesta dissertação de mestrado.

6.1 Conclusões e Contribuições

Esta pesquisa teve como contribuição, a proposição e validação experimental de uma abordagem de poda estruturada a partir do efeito causal entre neurônios de camadas completamente conectadas. A abordagem foi comparada com uma técnica mais comum, em que as conexões a serem podadas são escolhidas a partir da magnitude dos seus respectivos pesos, além de comparadas a partir de uma poda por meio de escolha aleatória dos neurônios a serem podados.

Sendo uma das primeiras pesquisas no contexto de poda em redes neurais profundas, a partir da análise causal entre os neurônios das camadas completamente conectadas, a presente pesquisa apresentou alguns resultados relevantes no Capítulo 5. Sendo possível concluir que os modelos resultantes após podas utilizando o efeito causal entre os neurônios apresentam ganhos de tempo e de espaço em disco, quando comparadas com o modelo sem poda. Isso acontece devido à abordagem proposta ser uma poda estrutural, onde neurônios e todas as suas conexões são removidas, caso esse neurônio seja escolhido na poda.

Com relação às acurácias apresentadas no Capítulo 5, o método proposto apresenta os melhores resultados, sem compartilhar o primeiro lugar (não há intersecção entre intervalos

de confiança para diferentes técnicas de poda, considerando o mesmo percentual de poda), em 3 níveis de poda (50%, 70% e 80%), e perde em outros 3 níveis de poda (40%, 60% e 90%), já para os outros 4 níveis de poda (10%, 20%, 30% e 95%), o método proposto compartilha o primeiro lugar, ou seja, há intersecção entre intervalos de confiança para diferentes técnicas de poda, considerando o mesmo percentual de poda. A poda por Magnitude e aleatória vencem em 1 nível, cada uma, 40% e 60%, respectivamente. Assim, no geral, a poda por ACE apresentou os melhores resultados de acurácia, de tempo de resposta e espaço de armazenamento em disco dos modelos.

6.2 Propostas para Trabalhos Futuros

Vislumbra-se como possível trabalho futuro a aplicação da abordagem de poda proposta em camadas convolucionais, visando reduzir a quantidade de filtros convolucionais e, consequentemente, reduzir tempo de execução e espaço em disco dos modelos. Ainda no contexto de poda, é possível adaptar a abordagem proposta para ser utilizada para selecionar conexões entre os neurônios que podem ser podados. Neste caso o resultado seria uma poda não estruturada, resultando em modelos esparsos, similarmente à abordagem por magnitude dos pesos que escolhe conexões com baixas magnitudes para serem podadas.

Outra aplicação futura do método proposto, seria usar a relação causa e efeito entre neurônios das camadas completamente conectadas com relação às saídas, na etapa da regularização *dropout*, ao se escolherem neurônios para serem desligados durante o treinamento, evitando-se assim uma escolha aleatória de neurônios.

Referências Bibliográficas

BLALOCK, Davis et al. What is the State of Neural Network Pruning? In: DHILLON, Inderjit S.; PAPAILIOPOULOS, Dimitris S.; SZE, Vivienne (Ed.). **Conference on Machine Learning and Systems (MLSys)**. [S.l.: s.n.], 2020. Disponível em: ¡http://dblp.uni-trier.de/db/conf/mlsys/mlsys2020.html#BlalockOFG20¿.

BROWN, Tom et al. Language Models are Few-Shot Learners. In: ADVANCES in Neural Information Processing Systems. [S.l.]: Curran Associates, Inc., 2020. v. 33, p. 1877–1901. Disponível em: ¡https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf¿.

CHATTOPADHYAY, Aditya et al. Neural Network Attributions: A Causal Perspective. In: PROCEEDINGS of the 36th International Conference on Machine Learning. [S.l.]: PMLR, 2019. v. 97, p. 981–990. Disponível em: ¡https://proceedings.mlr.press/v97/chattopadhyay19a.html¿.

DENG, Jia; DONG, Wei et al. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2009. P. 248–255. DOI: 10.1109/CVPR.2009.5206848.

DENG, Li. The mnist database of handwritten digit images for machine learning research. **IEEE Signal Processing Magazine**, IEEE, v. 29, n. 6, p. 141–142, 2012.

DENG, Li; YU, Dong. **Deep Learning: Methods and Applications**. [S.l.], mai. 2014. Disponível em: ¡https://www.microsoft.com/en-us/research/publication/deep-learning-methods-and-applications/¿.

EFRON, B. Bootstrap Methods: Another Look at the Jackknife. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 7, n. 1, p. 1–26, 1979. DOI: 10.1214/aos/1176344552. Disponível em: ¡https://doi.org/10.1214/aos/1176344552¿.

FERGUSON, Max et al. Automatic localization of casting defects with convolutional neural networks. In: 2017 IEEE International Conference on Big Data (Big Data). [S.l.: s.n.], 2017. P. 1726–1735. DOI: 10.1109/BigData.2017.8258115.

FRANKLE, Jonathan; CARBIN, Michael. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2019. Disponível em: ¡https://openreview.net/forum?id=rJl-b3RcF7¿.

GALE, Trevor; ELSEN, Erich; HOOKER, Sara. The State of Sparsity in Deep Neural Networks. **Computing Research Repository (CoRR)**, abs/1902.09574, 2019. arXiv: 1902.09574. Disponível em: ¡http://arxiv.org/abs/1902.09574¿.

GOUDET, Olivier et al. Causal Generative Neural Networks, 2017. cite arxiv:1711.08936. Disponível em: ¡http://arxiv.org/abs/1711.08936¿.

HAN, Song; MAO, Huizi; DALLY, William. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In.

HAN, Song; POOL, Jeff et al. Learning both Weights and Connections for Efficient Neural Networks. **Computing Research Repository (CoRR)**, abs/1506.02626, 2015. arXiv: 1506.02626. Disponível em: ¡http://arxiv.org/abs/1506.02626¿.

HARRADON, Michael; DRUCE, Jeff; RUTTENBERG, Brian E. Causal Learning and Explanation of Deep Neural Networks via Autoencoded Activations. **Computing Research Repository (CoRR)**, abs/1802.00541, 2018. arXiv: 1802.00541. Disponível em: ¡http://arxiv.org/abs/1802.00541¿.

HASSIBI, Babak; STORK, David. Second order derivatives for network pruning: Optimal Brain Surgeon. In: ADVANCES in Neural Information Processing Systems. [S.l.]: Morgan-Kaufmann, 1992. v. 5. Disponível em:

¡https://proceedings.neurips.cc/paper_files/paper/1992/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf¿.

HE, Kaiming et al. Deep Residual Learning for Image Recognition. **Computing Research Repository (CoRR)**, abs/1512.03385, 2015. arXiv: 1512.03385. Disponível em: ¡http://arxiv.org/abs/1512.03385¿.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks. In: PEREIRA, F. et al. (Ed.). **Advances in Neural Information Processing Systems 25**. [S.l.]: Curran Associates, Inc., 2012. P. 1097–1105. Disponível em: ¡http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf¿.

LECUN, Yann; DENKER, John; SOLLA, Sara. Optimal brain damage. **Advances in neural information processing systems**, v. 2, 1989.

LIU, Weibo et al. A survey of deep neural network architectures and their applications. **Neurocomputing**, v. 234, dez. 2016. DOI: 10.1016/j.neucom.2016.12.038.

LIU, Zhuang et al. Rethinking the Value of Network Pruning. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2019. Disponível em: ¡https://openreview.net/forum?id=rJlnB3C5Ym¿.

NAIR, Vinod; HINTON, Geoffrey E. Rectified linear units improve restricted boltzmann machines. In: ICML 2010. [S.l.: s.n.], 2010. P. 807–814.

O'SHEA, Keiron; NASH, Ryan. An Introduction to Convolutional Neural Networks. **Computing Research Repository (CoRR)**, abs/1511.08458, 2015. arXiv: 1511.08458. Disponível em: ¡http://arxiv.org/abs/1511.08458¿.

PEARL, J.; GLYMOUR, M.; JEWELL, N.P. **Causal Inference in Statistics: A Primer**. [S.l.]: Wiley, 2016. ISBN 9781119186847. Disponível em: ¡https://books.google.com.br/books?id=L3G-CgAAQBAJ¿.

PEARL, Judea. **Causality**: Models, Reasoning, and Inference. 2. ed. [S.l.]: Cambridge University Press, 2009. ISBN 978-0-521-89560-6. DOI: 10.1017/CBO9780511803161.

RUSSAKOVSKY, Olga et al. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

SIMONYAN, Karen; ZISSERMAN, Andrew. Very Deep Convolutional Networks for Large-Scale Image Recognition. **Computing Research Repository (CoRR)**, abs/1409.1556, 2014. Disponível em: ¡http://arxiv.org/abs/1409.1556¿.

SUN, Xudong; WU, Pengcheng; HOI, Steven C. H. Face Detection using Deep Learning: An Improved Faster RCNN Approach. **Computing Research Repository (CoRR)**, abs/1701.08289, 2017. arXiv: 1701.08289. Disponível em: ¡http://arxiv.org/abs/1701.08289¿.

SZEGEDY, Christian et al. Going Deeper with Convolutions. In: COMPUTER Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2015. Disponível em: ¡http://arxiv.org/abs/1409.4842¿.

WANG, H. et al. Structured Pruning for Efficient Convolutional Neural Networks via Incremental Regularization. **IEEE Journal of Selected Topics in Signal Processing**, p. 1–1, 2019.

XIN YAO. Evolving artificial neural networks. **Proceedings of the IEEE**, v. 87, n. 9, p. 1423–1447, 1999.

XUE, Hongyang et al. Tracking people in RGBD videos using deep learning and motion clues. **Neurocomputing**, v. 204, abr. 2016. DOI: 10.1016/j.neucom.2015.06.112.

YEOM, Seul-Ki et al. Pruning by Explaining: A Novel Criterion for Deep Neural Network Pruning. **Computing Research Repository (CoRR)**, abs/1912.08881, 2019. arXiv: 1912.08881. Disponível em: ¡http://arxiv.org/abs/1912.08881¿.

YU, Ruichi et al. NISP: Pruning Networks using Neuron Importance Score Propagation. **Computing Research Repository (CoRR)**, abs/1711.05908, 2017. arXiv: 1711.05908. Disponível em: ¡http://arxiv.org/abs/1711.05908¿.

ZHANG, J. et al. SNAP: A 1.67 — 21.55TOPS/W Sparse Neural Acceleration Processor for Unstructured Sparse Deep Neural Network Inference in 16nm CMOS. In: 2019 Symposium on VLSI Circuits. [S.l.: s.n.], 2019. P. c306–c307.

Apêndice A

Exemplo de cálculo do ACE

Considere o seguinte exemplo: um estudo de um novo medicamento para o tratamento de uma doença foi realizado. Esse estudo considerou 700 pacientes para a avalização da eficácia, ou não, desse medicamento. Os pacientes foram divididos em dois grupos: 1) 350 pacientes receberam o medicamento; 2) 350 pacientes não receberam o medicamento. Na Tabela A.1, são apresentados os resultados obtidos com o estudo, onde a primeira linha corresponde aos pacientes do sexo masculino, a segunda aos pacientes do sexo feminino e a terceira aos resultados de todos os pacientes.

Tabela A.1: Resultados dos estudos de um novo medicamento. Exemplo adaptado de Pearl, Glymour e Jewell (2016).

	Tomou o medicamento	**Não tomou o medicamento**
Homens	81 de 87 recuperados (93%)	234 de 270 recuperados (87%)
Mulheres	192 de 263 recuperados (73%)	55 de 80 recuperados (69%)
Dados combinados	273 de 350 recuperados (78%)	289 de 350 recuperados (83%)

A partir da Tabela A.1, é possível perceber que tanto para os homens quanto para as mulheres, tomar o medicamento ajuda mais na recuperação do que não tomar o medicamento, onde a medicação ajudou em 93% dos casos dos pacientes homens, contra 87% dos homens que não tomaram o medicamento foram recuperados. Já no contexto das mulheres, houve 73% de recuperações para as mulheres que tomaram o medicamento, contra 60% de recuperações para as mulheres que não tomaram o medicamento. Contudo, pode-se observar

que ao considerar os dados combinados, não tomar o medicamento traz melhores resultados do que tomar, com 83% de recuperados contra 78% de recuperados, respectivamente. Mas afinal, a medicação ajuda ou não ajuda no tratamento da doença?

Neste problema, tem-se o paradoxo de Simpson, como apresentado na Seção 2.3, em que ocorrem interpretações opostas ao analisar os dados de forma global ou agrupada. Esse problema pode ser resolvido a partir do cálculo do ACE, aplicando a Equação 4.2, considerando *X = 1* como paciente que tomou a medicação, *Z = 1* como pacientes do gênero masculino, *Y = 1* como o paciente se recuperou da doença, *X = 0* como paciente que não tomou a medicação, *Z = 0* como pacientes do gênero feminino e *Y = 0* como o paciente que não se recuperou da doença, tem-se:

$P(Y = 1|do(X = 1)) = P(Y = 1|X = 1, Z = 1)P(Z = 1) + P(Y = 1|X = 1, Z = 0)P(Z = 0)$ e

$P(Y = 1|do(X = 0)) = P(Y = 1|X = 0, Z = 1)P(Z = 1) + P(Y = 1|X = 0, Z = 0)P(Z = 0)$

O grafo gerado a partir do problema, é apresentado na Figura A.1, onde é já é considerada a intervenção em *X*, como na demonstração acima.

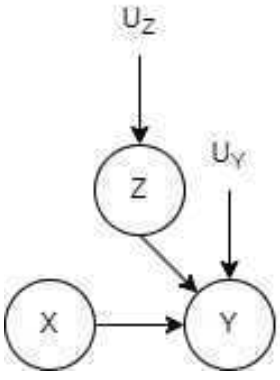


Figura A.1: Grafo representando a relação entre gênero (Z), tomou a medicação (X) e recuperação da doença (Y). Fonte: Adaptado de Pearl, Glymour e Jewell (2016).

A partir dos dados apresentados na Tabela A.1, tem-se:

$$P(Y = 1|do(X = 1)) = \frac{0,93 \times (87 + 270)}{700} + \frac{0,73 \times (263 + 80)}{700} = 0,832$$

$$P(Y = 1|do(X = 0)) = \frac{0,87 \times (87 + 270)}{700} + \frac{0,69 \times (263 + 80)}{700} = 0,7818$$

Logo, o valor do ACE será o seguinte:

$$ACE = P(Y = 1|do(X = 1)) - P(Y = 1|do(X = 0)) = 0,832 - 0,7818 = 0,0502$$

Assim, como o valor do ACE é um número maior que zero ($ACE > 0$), pode-se concluir que tomar o medicamento é benéfico no tratamento da doença.

Apêndice B

Descrição das classes da base de dados utilizadas

Tabela B.1: Classes de imagens da ImageNet, bem como seus respectivos códigos e uma descrição, utilizadas na presente pesquisa

Código	**Classe**	**Descrição**
n01440764	Pássaro canoro	Pássaros conhecidos por sua habilidade de cantar e muitas vezes criados como animais de estimação.
n01514859	Ganso-do-canadá	Ave aquática de grande porte que se alimenta principalmente de plantas e pode ser encontrada em várias partes do mundo.
n01560419	Galo-da-serra	Ave encontrada nas montanhas da América do Sul, conhecida por suas penas brilhantes e pela habilidade de imitar sons de outros animais.
n01622779	Tartaruga-de-casco-mole	Espécie de tartaruga com um casco mole que pode ser encontrado em água doce ou salgada em várias partes do mundo.

n01644900	Esturjão	Peixe de água doce ou salgada encontrado em várias partes do mundo, conhecido por seu valor culinário e uso na produção de caviar.
n01682714	Colibri	Pequena ave encontrada nas Américas conhecida por suas habilidades de voo e cores brilhantes.
n01695060	Mantis-religiosa	Inseto predador comum em várias partes do mundo, conhecido por sua postura característica e habilidades de caça.
n01443537	Pica-pau	Pássaro conhecido por sua habilidade de perfurar madeira com o bico para encontrar alimento ou construir um ninho.
n01518878	Galinha	Ave domesticada comum em várias partes do mundo, criada principalmente para produção de ovos ou carne.
n01580077	Beija-flor	Pequena ave encontrada nas Américas conhecida por suas habilidades de voo e cores brilhantes.
n01629819	Pagode	Um tipo de templo budista que é característico da arquitetura chinesa e japonesa.
n01664065	Tartaruga de casco mole	Uma tartaruga que não tem um casco rígido como as outras tartarugas, e é capaz de dobrá-lo e movimentá-lo com maior facilidade.

n01685808	Caracol	Um animal invertebrado que possui uma concha espiralada nas costas e se movimenta lentamente.
n01697457	Tartaruga de casco duro	Uma tartaruga que tem um casco rígido e é encontrado em várias partes do mundo.
n01484850	Narceja-comum	Uma ave aquática pequena e migratória, com bico longo e fino.
n01530575	Águia-de-cabeça-branca-americana	Uma ave de rapina encontrada na América do Norte, com uma plumagem predominantemente marrom e cabeça branca.
n01582220	Cardeal	Uma ave que é comum na América do Norte e tem plumagem vermelha brilhante na cabeça e no peito.
n01630670	Lagarto-monstro de Gila	Um lagarto venenoso encontrado no deserto americano, com manchas escuras na pele.
n01665541	Tartaruga-marinha	Uma tartaruga encontrada em águas marinhas em todo o mundo, que é ameaçada de extinção.
n01687978	Ouriço-cacheiro	Um pequeno animal coberto de espinhos, encontrado em várias partes do mundo.
n01491361	cágado	réptil aquático com casco e patas para nado
n01531178	avestruz	grande ave corredora, incapaz de voar
n01592084	raposa-vermelha	mamífero carnívoro de porte médio com pelagem vermelha

n01631663	cascavel	serpente venenosa com um chocalho na ponta da cauda
n01667114	tartaruga-verde	réptil marinho ameaçado de extinção
n01688243	caranguejo-ferradura	artrópode marinho com formato de ferradura e considerado um fóssil vivo
n01494475	tenrec	mamífero exótico, originário de Madagascar, que lembra um ouriço
n01532829	salamandra	anfíbio com quatro patas, cauda e pele úmida
n01601694	urso-pardo	grande mamífero carnívoro com pelagem marrom-escuro
n01632458	iguana-verde	réptil arbóreo comum em países tropicais
n01667778	*Chelonia mydas*	Tartaruga-verde, espécie de tartaruga marinha encontrada em todo o mundo.
n01689811	*Balaenoptera acutorostrata*	Baleia-de-minke-anã, espécie de baleia encontrada em todos os oceanos do mundo.
n01496331	*Ardea cinerea*	Garça-real, espécie de ave encontrada em grande parte da Europa e Ásia.
n01534433	*Pavo cristatus*	Pavão, espécie de ave da família Phasianidae, nativa da Ásia.
n01608432	*Scolopax rusticola*	Narceja-comum, espécie de ave da família Scolopacidae, nativa da Europa, Ásia e norte da África.
n01632777	*Ursus americanus*	Urso-negro-americano, espécie de urso encontrado na América do Norte.
n01669191	*Fulica atra*	Galeirão, espécie de ave da família Rallidae, nativa da Europa, Ásia e norte da África.

n01692333	*Equus caballus*	Cavalo, espécie de mamífero da família Equidae, domesticado pelo homem.
n01498041	*Turdus merula*	Melro-preto, espécie de ave da família Turdidae, nativa da Europa, Ásia e norte da África.
n01537544	*Threskiornis aethiopicus*	Ibis-sagrado, espécie de ave da família Threskiornithidae, nativa da África e Ásia.
n01614925	Orangotango	Esta classe representa a espécie de grandes macacos nativos da Ásia, conhecidos por sua pelagem avermelhada e braços longos.
n01641577	Tartaruga verde	Esta classe representa uma espécie de tartaruga marinha comumente encontrada em águas tropicais e subtropicais.
n01675722	Víbora-das-árvores-verde	Esta classe representa uma espécie de cobra venenosa, encontrada em áreas de florestas tropicais na América Central e do Sul.
n01693334	Porco-espinho	Esta classe representa um animal roedor coberto de espinhos pontiagudos que se defende de predadores, encontrados em todo o mundo.
n01514668	Gaivota-prateada	Esta classe representa uma espécie de ave marinha com plumagem predominantemente branca e cinza, encontrada em todo o mundo.

n01558993	Cardeal	Esta classe representa uma espécie de ave de bico grosso e coloração vermelha brilhante encontrada nas Américas do Norte e do Sul.
n01616318	Cágado	Esta classe representa uma espécie de tartaruga terrestre encontrada em todo o mundo, conhecida por sua carapaça convexa.
n01644373	Cobra-rei	Esta classe representa uma espécie de cobra venenosa encontrada no sudeste asiático e na Índia, conhecida por sua coloração brilhante e agressividade.
n01677366	Aranha caranguejeira	Esta classe representa uma espécie de aranha grande e peluda, encontrada nas florestas tropicais da América do Sul.
n01694178	Águia-cinzenta	Esta classe representa uma espécie de águia nativa da América do Norte, conhecida por sua plumagem cinza-escuro.

Apêndice C

Definição do limiar para ativação dos neurônios

Para computar as ativações dos neurônios foram utilizadas as informações de média e desvio padrão, conforme descrito na Subseção 4.3. Assim, um neurônio é considerado ativado quando seu valor estiver em pelo menos dois desvios padrões acima da média. Para a decisão do limiar 2, fez-se necessário a realização de alguns experimentos, variando esse limiar. Assim, a Tabela C.1 apresenta os resultados obtidos a partir da variação do limiar de ativação dos neurônios. Onde, top_n indica a acurácia obtida ao considerar as n classe mais bem ranqueadas, ou seja, se a classe esperada estiver dentro das n classes mais bem ranqueadas, considera-se um acerto. Já os valores em negrito, indicam o melhor resultado ao considerar o percentual de poda aplicado. Assim, considerando o limiar de ativação 2, é possível perceber que ele apresenta os melhores resultados.

Tabela C.1: Resultados obtidos a partir da variação do limiar de ativação dos neurônios

Limiar de ativação	**Percentual de Poda**	**Acurácia**				
		top_1	**top_2**	**top_3**	**top_4**	**top_5**
1	0,1	**0,8234**	**0,9085**	**0,9382**	0,9486	0,9559
	0,2	0,8250	**0,9149**	0,9382	0,9526	**0,9639**
	0,3	0,8234	0,9101	0,9446	0,9567	0,9647
	0,4	0,8090	0,9109	0,9374	**0,9575**	0,9663
	0,5	0,7913	0,8965	0,9278	0,9478	0,9583
	0,7	0,6461	0,7929	0,8475	0,8772	0,8957
	0,8	0,5040	0,6653	0,7239	0,7713	0,7913
2	0,1	0,8226	0,9045	0,9374	**0,9502**	0,9575
	0,2	**0,8266**	0,9117	0,9374	0,9535	0,9615
	0,3	0,8242	0,9085	0,9438	**0,9583**	0,9631
	0,4	**0,8210**	**0,9125**	**0,9422**	0,9559	0,9647
	0,5	**0,8066**	**0,9085**	**0,9382**	**0,9543**	**0,9647**
	0,7	0,7071	0,8283	0,8804	0,8989	0,9165
	0,8	**0,5915**	**0,7424**	**0,8090**	**0,8451**	**0,8652**
3	0,1	0,8218	0,9069	0,9358	0,9486	0,9583
	0,2	0,8218	0,9085	0,9390	0,9518	0,9623
	0,3	**0,8274**	0,9125	0,9430	0,9543	0,9631
	0,4	0,8098	0,9037	0,9374	0,9559	0,9655
	0,5	0,7986	0,8973	0,9334	0,9526	0,9615
	0,7	0,7022	0,8315	0,8748	0,8997	0,9149
	0,8	0,5843	0,7319	0,7945	0,8250	0,8507
4	0,1	0,8226	0,9061	0,9366	0,9486	**0,9591**
	0,2	0,8258	0,9085	0,9374	**0,9543**	0,9631
	0,3	0,8250	**0,9133**	0,9406	0,9551	0,9647
	0,4	0,8202	0,9101	**0,9422**	**0,9575**	**0,9671**
	0,5	0,8018	0,9045	0,9358	0,9535	0,9623
	0,7	**0,7175**	**0,8363**	**0,8828**	**0,9053**	**0,9205**

	0,8	0,5899	0,7360	0,8010	0,8331	0,8612
5	0,1	0,8226	0,9077	0,9374	0,9486	0,9567
	0,2	0,8234	0,9093	**0,9406**	0,9502	0,9631
	0,3	0,8194	0,9117	**0,9454**	0,9543	**0,9655**
	0,4	0,8106	0,9061	0,9390	**0,9575**	0,9655
	0,5	0,7897	0,8925	0,9270	0,9494	0,9607
	0,7	0,6709	0,8210	0,8764	0,8973	0,9165
	0,8	0,5465	0,7047	0,7785	0,8098	0,8331