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

WHENDELL FEIJÓ MAGALHÃES

PODA ESTRUTURADA DE REDES NEURAIS CONVOLUCIONAIS E A HIPÓTESE DO BILHETE DE LOTERIA

CAMPINA GRANDE - PB
2021

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

Poda Estruturada de Redes Neurais Convolucionais e a Hipótese do Bilhete de Loteria

Whendell Feijó Magalhães

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: Compressão de Redes Neurais

Herman Martins Gomes & Leandro Balby Marinho
(Orientadores)

Campina Grande, Paraíba, Brasil

M188p Magalhães, Whendell Feijó.

Poda estruturada de redes neurais convolucionais e a hipótese do bilhete de loteria / Whendell Feijó Magalhães. – Campina Grande, 2022.

77 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2021.

"Orientação: Prof. Dr. Herman Martins Gomes; Coorientação: Prof. Dr. Leandro Balby Marinho".

Referências.

1. Aprendizagem Profunda. 2. Compressão de Redes Neurais Convolucionais. 3. Poda Estruturada. 4. Explicabilidade de Redes Neurais. 5. Ciência da Computação. I. Gomes, Herman Martins. II. Marinho, Leandro Balby. III. Título.

CDU 004.032.26(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA MARIA ANTONIA DE SOUSA CRB 15/398

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO CIENCIAS DA COMPUTACAO
Rua Aprigio Veloso, 882, - Bairro Universitario, Campina Grande/PB, CEP 58429-900

FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES

WHENDELL FEIJÓ MAGALHÃES

PODA ESTRUTURADA DE REDES NEURAIS CONVOLUCIONAIS E A HIPÓTESE DO BILHETE DE LOTERIA

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: 16/12/2021

Prof. Dr. HERMAN MARTINS GOMES, Orientador, UFCG

Prof. Dr. LEANDRO BALBY MARINHO, Orientador, UFCG

Prof. Dr. EANES TORRES PEREIRA, Examinador Interno, UFCG

Prof. Dr. ADRIANO ALONSO VELOSO, Examinador Externo, UFMG

Documento assinado eletronicamente por **LEANDRO BALBY MARINHO**, **PROFESSOR 3 GRAU**, em 17/12/2021, às 08:10, conforme horário oficial de Brasília, com fundamento no art. 8°, caput, da Portaria SEI n° 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **HERMAN MARTINS GOMES**, **PROFESSOR 3 GRAU**, em 19/12/2021, às 20:18, conforme horário oficial de Brasília, com fundamento no art. 8°, caput, da Portaria SEI n° 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **EANES TORRES PEREIRA**, **PROFESSOR 3 GRAU**, em 20/12/2021, às 08:21, conforme horário oficial de Brasília, com fundamento no art. 8°, caput, da Portaria SEI n° 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 **2014699** e o código CRC **CCC4FAFD**.

Resumo

A Hipótese do Bilhete de Loteria formula que é possível encontrar sub-redes (bilhetes vencedores) que apresentam acurácia igual ou superior à rede não podada e alta capacidade de generalização, quando obtida a partir de uma rede neural super-parametrizada. Uma etapa do algoritmo que implementa a hipótese requer o rebobinamento dos pesos da rede podada para seus valores iniciais, normalmente valores aleatórios. Variações mais recentes dessa etapa podem envolver: (i) redefinir os pesos para os valores que eles tinham em uma época inicial do treinamento da rede não podada (rebobinamento dos pesos), ou (ii) manter os pesos finais do treinamento e redefinir apenas a taxa de aprendizado (rebobinamento da taxa de aprendizagem). Apesar de algumas pesquisas terem investigado as variações acima, a maioria em poda não estruturada (poda de pesos), não há, com base na revisão bibliográfica desta pesquisa, avaliações existentes focadas em poda estruturada (poda de neurônios ou filtros) para as variantes de poda local e global. Além disso, as pesquisas relacionadas à hipótese do bilhete de loteria utilizam somente a magnitude dos pesos como critério de seleção dos elementos a serem podados. Neste contexto, esta pesquisa apresenta novas evidências empíricas de que é possível obter bilhetes vencedores ao realizar a poda estruturada de redes neurais convolucionais e propõe a utilização de um critério de poda baseado na técnica de explicabilidade DeepLIFT como alternativa à magnitude dos pesos. Para isso, configurou-se um experimento utilizando a rede VGG16 treinada nos conjuntos de dados CIFAR-10 e CIFAR-100 e comparou-se com redes (podadas em diferentes níveis de compressão) obtidas pelos métodos de rebobinamento dos pesos e rebobinamento da taxa de aprendizagem, nos contextos de poda local (orientada à camada) e poda global (independente da camada). Usou-se a rede não podada como base para as comparações e também comparou-se as redes podadas resultantes com suas versões treinadas com pesos inicializados aleatoriamente. Além disso, ainda avaliou-se o impacto da substituição da magnitude dos pesos pelo método DeepLIFT em redes podadas de forma global com a abordagem de rebobinamento da taxa de aprendizagem. De modo geral, ao utilizar a poda global, o rebobinamento dos pesos produziu alguns bilhetes vencedores (limitados a baixos níveis de poda) e com desempenho igual ou pior em comparação com a inicialização aleatória. O rebobinamento da taxa de

aprendizagem, ao utilizar a poda global, produziu os melhores resultados dentre as abordagens de rebobinamento, uma vez que encontrou bilhetes vencedores em diferentes níveis de poda, inclusive para níveis mais agressivos. Além disso, as redes podadas usando o método DeepLIFT como critério de poda, ao final das iterações de poda, apresentaram acurácia média maior que as redes podadas usando a magnitude dos pesos, além de maior estabilidade e tolerância a níveis de poda mais agressivos. Por fim, foi possível verificar uma redução significativa no tempo de inferência (*speedup* de $\approx 5\times$ em *batches* de tamanho 1 e de $\approx 4\times$ em *batches* de tamanho 128) das redes podadas quando executadas em CPU, produzindo assim redes mais adequadas à execução em dispositivos com poucos recursos computacionais.

Abstract

The Lottery Ticket Hypothesis formulates that it is possible to find subnetworks (winning tickets) that has the same or higher accuracy than the unpruned and high generalization capabilities, if obtained from an over-parameterized neural network. One step of the algorithm implementing the hypothesis requires resetting the weights of the pruned network to their initial random values. More recent variations of this step may involve: (i) resetting the weights to the values they had at an early epoch of the unpruned network training (weight rewinding), or (ii) keeping the final training weights and resetting only the learning rate schedule (learning rate rewinding). Despite some studies have investigated the above variations, mostly with unstructured pruning (weight pruning), we do not know of existing evaluations focusing on structured pruning (neuron pruning or filter pruning) regarding local and global pruning variations. Furthermore, studies related to the lottery ticket hypothesis uses only the magnitude of the weights as criteria for selecting the elements to be pruned. In this context, this research presents novel empirical evidence that it is possible to obtain winning tickets when performing structured pruning of convolutional neural networks and proposes the use of a pruning criteria based on the DeepLIFT explainability technique as an alternative to weights magnitude. We setup an experiment using the VGG16 network trained on the CIFAR-10 and CIFAR-100 datasets and compared with networks (pruned at different compression levels) got by weight rewinding and learning rate rewinding methods, under local (layer-wise) and global (layer-independent) pruning regimes. The unpruned network was used as baseline and also compared the resulting pruned networks with their versions trained with randomly initialized weights. Furthermore, the impact of replacing the magnitude of the weights with the DeepLIFT method on globally pruned networks with the learning rate rewinding method was evaluated. Overall, when using global pruning, rewinding the weights produced a few winning tickets (limited to low levels of pruning) and with equal or worse performance compared to random initialization. Learning rate rewinding, when using global pruning, weight rewinding produced a few winning tickets (limited to low pruning levels only) and performed nearly the same or worse compared to random initialization. Learning rate rewinding, under global pruning, produced the best results, since it has found winning tickets at different pruning levels, even for more aggressive levels. Furthermore, networks pruned using the

DeepLIFT method as pruning criteria, at the end of pruning iterations, presented higher average accuracy than networks pruned using the magnitude of weights, as well as higher stability and tolerance to more aggressive pruning levels. Finally, a significant reduction in inference time ($\approx 5\times$ speedup on batches of size 1 and $\approx 4\times$ speedup on batches of size 128) of the pruned networks when run on CPU could be verified, thus resulting in networks more suitable for execution on devices with low computational resources.

Agradecimentos

Em primeiro lugar gostaria de agradecer aos meus orientadores, Herman Martins Gomes e Leandro Balby Marinho, pela parceria, paciência e orientação durante toda a trajetória do mestrado.

Agradeço imensamente à minha companheira Júlia Farias, que sempre foi meu porto seguro ao longo de toda essa trajetória. Agradeço também aos meus pais, Adriana Feijó e Wilton Ribeiro, por sempre me incentivarem a estudar, reforçando a importância da educação.

Por fim agradeço aos colegas, professores e demais servidores e funcionários que fazem da UFCG este ambiente propício à prática acadêmica, onde tive oportunidades incríveis e que com certeza mudaram minha vida.

Conteúdo

Lista de Abreviaturas e Siglas

CNN - *Convolutional Neural Network* ou *Rede Neural Convolucional*

IMP - *Iterative Magnitude Pruning* ou *Poda Iterativa Baseada em Magnitude*

SGD - *Stochastic Gradient Descent* ou *Gradiente Descendente Estocástico*

XAI - *Explainable Artificial Intelligence* ou *Inteligência Artificial Explicável*

LRP - *Layer-Wise Relevance Propagation*

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Capítulo 1

Introdução

Soluções baseadas em Inteligência Artificial são uma realidade cada vez mais frequente em nosso cotidiano. Isso se deve em boa parte aos avanços alcançados em Aprendizagem Profunda (*Deep Learning*), com destaque para as Redes Neurais Profundas (*Deep Neural Networks*) (LECUN; BENGIO; HINTON, 2015).

Redes Neurais Profundas são o estado da arte para diversas tarefas complexas baseadas em IA, como visão computacional, processamento de linguagem natural e reconhecimento da fala, dentre outras (LIU, W. et al., 2017). Como exemplos de aplicações que utilizam redes neurais profundas nós temos carros autônomos, segurança em cidades inteligentes, diagnóstico assistido por computador, análise de sentimentos e *chatbots*.

No entanto, a execução de redes neurais profundas não é uma tarefa trivial por conta do alto custo computacional tanto para treinamento em grandes conjuntos de dados, quanto para inferência, por conta do número de parâmetros e operações executadas. Por conta dessa característica, a execução dessas redes está geralmente associada a hardware especializado, tais como unidades de processamento gráfico (*Graphics Processing Units*) ou ainda unidades de processamento de tensores (*Tensor Processing Units*).

Devido à natureza ávida de recursos das redes neurais profundas, sua execução em computadores com recursos limitados, como, por exemplo, dispositivos de borda e computadores de prateleira, é muitas vezes inviável. Por conta disso, a otimização de redes neurais se torna cada vez mais importante, pois tenta viabilizar a execução destas redes em hardware menos robusto e em última instância, democratiza o acesso a estas soluções.

1.1 Motivação

Apesar dos resultados excepcionais alcançados, as arquiteturas das redes neurais profundas são complexas e estão em constante evolução. Essas redes, em sua maioria, são super-parametrizadas (BA; CARUANA, 2014; DU; LEE, 2018; LI; LIANG, 2018). Por super-parametrizadas, entende-se as redes neurais profundas em que o número de parâmetros é alto e por vezes até supera o número de características de entrada ou o número de instâncias de treinamento. Como consequência, essas redes apresentam alto custo computacional tanto para treinamento quanto para inferência.

Para fins de comparação, LeNet-5, uma das primeiras redes neurais convolucionais, introduzida por LeCun et al. (1998), possui aproximadamente 60 mil parâmetros e foi desenvolvida para classificação do conjunto de dados MNIST que possui 60 mil imagens em escala de cinza de 28×28 pixels. Enquanto isso, redes neurais de maior profundidade desenvolvidas mais recentemente alcançam facilmente centenas de milhões de parâmetros, como a arquitetura VGG-16 introduzida por Simonyan & Zisserman (2015) (SIMONYAN; ZISSERMAN, 2015), que possui aproximadamente 138 milhões de parâmetros e foi projetada para classificação do desafio ILSVRC-2014 (RUSSAKOVSKY et al., 2015), que tem 1 milhão de imagens coloridas de treinamento com resolução média de 482×415 pixels.

Apesar da super-parametrização ser um inconveniente por implicar aumento do custo computacional das redes neurais profundas, vale notar que tal característica é bem conhecida por ajudar os algoritmos de busca locais a alcançar baixo erro no treinamento, uma vez que favorece a criação de uma grande variedade de soluções ótimas durante o treinamento (LIVNI; SHALEV-SHWARTZ; SHAMIR, 2014). Além disso, a super-parametrização, combinada com regularização, pode ajudar a obter melhor generalização, ou seja, aplicar o que foi aprendido durante o treinamento em dados nunca antes vistos.

Dado o considerável espaço de busca envolvido na construção de arquiteturas DNN sob medida para tarefas específicas, é prática comum utilizar arquiteturas *off-the-shelf*, bem conhecidas por operarem bem em várias tarefas complexas. Se o modelo selecionado for mais complexo do que o necessário, pode-se aplicar técnicas para reduzir a complexidade do modelo (por exemplo, regularização), mantendo-se (ou mesmo melhorando em alguns casos) o poder de generalização do modelo original.

Por conta disso, recentemente, muitas pesquisas têm surgido propondo novas técnicas para permitir o treinamento e execução destas redes em hardware de uso geral, uma área de pesquisa emergente que é comumente chamada de compressão de redes neurais (CHOUDHARY et al., 2020). Dentre as técnicas de compressão existentes, a poda de rede neural é uma categoria bem estabelecida de técnicas que visa reduzir o consumo de armazenamento, de memória e o uso de recursos computacionais, sem prejudicar significativamente a acurácia das redes neurais profundas, mesmo quando podadas em níveis muito agressivos (FRANKLE; CARBIN, 2019).

No contexto de poda de redes neurais, pode-se citar as seguintes variações: Poda Não-Estruturada, Poda Estruturada, Poda Local e Poda Global. A poda não estruturada, ou poda de pesos, consiste na remoção de conexões entre neurônios e/ou filtros convolucionais de camadas adjacentes. Já a poda estruturada, consiste na remoção de estruturas inteiras da rede neural, que podem ser neurônios, filtros convolucionais ou até mesmo camadas inteiras. Quando aplica-se Poda Local, os neurônios e/ou filtros convolucionais são removidos de forma proporcional em todas as camadas da rede neural. Enquanto na Poda Global, os neurônios e/ou filtros convolucionais são removidos em diferentes proporções por camada, sem levar em consideração a camada em que a poda está sendo aplicada. Maiores detalhes sobre os diferentes aspectos relacionados à poda de redes neurais são apresentados na Seção 2.2.

Pesquisas recentes demonstraram que através do retreinamento total de uma rede podada é possível atingir a mesma acurácia e por vezes até superar sua contraparte não podada. Tais pesquisas se referem a essas redes podadas que igualam ou até superam a acurácia das redes não podadas como bilhetes vencedores (FRANKLE; CARBIN, 2019; FRANKLE et al., 2019; MORCOS et al., 2019; ZHOU, H. et al., 2019; RENDA; FRANKLE; CARBIN, 2020) em referência à Hipótese do Bilhete de Loteria, que é apresentada em mais detalhes na Seção 3.1. Nessas pesquisas, os autores propõem e/ou utilizam duas abordagens durante o retreinamento das redes podadas, sendo elas Rebobinamento dos Pesos (*Weight Rewinding*) e Rebobinamento da Taxa de Aprendizagem (*Learning Rate Rewinding*). A abordagem de Rebobinamento dos Pesos consiste em reiniciar os pesos da rede podada para os valores que tinham em uma das época iniciais do treinamento da rede não podada antes de realizar o retreinamento (FRANKLE; CARBIN, 2019; FRANKLE et al., 2019; MORCOS et al., 2019;

ZHOU, H. et al., 2019). Já na abordagem de Rebobinamento da Taxa de Aprendizagem, os pesos do treinamento original e, posteriormente, das iterações de retreinamento são mantidos na rede podada e somente o protocolo de variação da taxa de aprendizagem é redefinido antes de realizar o retreinamento (RENDA; FRANKLE; CARBIN, 2020).

Entretanto, foram identificadas duas limitações recorrentes em pesquisas sobre poda de redes neurais profundas relacionadas à Hipótese do Bilhete de Loteria:

1. A maioria das descobertas e conclusões são baseadas em experimentos com poda não-estruturada (FRANKLE; CARBIN, 2019; FRANKLE et al., 2019; MORCOS et al., 2019; ZHOU, H. et al., 2019; RENDA; FRANKLE; CARBIN, 2020);

2. A seleção dos elementos a serem podados é feita com base na magnitude dos pesos (FRANKLE; CARBIN, 2019; FRANKLE et al., 2019; MORCOS et al., 2019; RENDA; FRANKLE; CARBIN, 2020).

A utilização da magnitude dos pesos se baseia no fato de que os filtros que possuem as menores magnitudes resultam em menores valores de ativação e portanto têm menor impacto na formação da predição da rede neural (HE, Y. et al., 2018). Entretanto, apesar de pesquisas recentes demonstrarem a efetividade da remoção de elementos com base na magnitude dos pesos (HAN et al., 2015; LI et al., 2017; FRANKLE; CARBIN, 2019; HE, Y. et al., 2019; ZHOU, H. et al., 2019), a relação direta entre magnitude e importância dos pesos é questionada desde pesquisas seminais da literatura de poda de redes neurais devido à sua natureza empírica (LECUN; DENKER; SOLLA, 1990; HASSIBI; STORK, 1993).

Uma área que vem se tornando cada vez mais popular e relevante é a área de Inteligência Artificial Explicável, uma vez que com a adoção massiva de soluções de inteligência artificial surge a necessidade de prover mais transparência para essas soluções (MILLER, 2019). As pesquisas dessa área têm como objetivo prover mecanismos que nos permita entender a lógica por trás das decisões tomadas pelos algoritmos de inteligência artificial. No que diz respeito especificamente às redes neurais profundas tem-se a sub-área de Explicabilidade de Redes Neurais Profundas.

Em geral, os métodos de explicabilidade de redes neurais profundas funcionam atribuindo pontuações de importância - por vezes chamados de contribuição - a cada uma das variáveis de entrada da rede neural (BACH et al., 2015; MONTAVON, Grégoire et al., 2017;

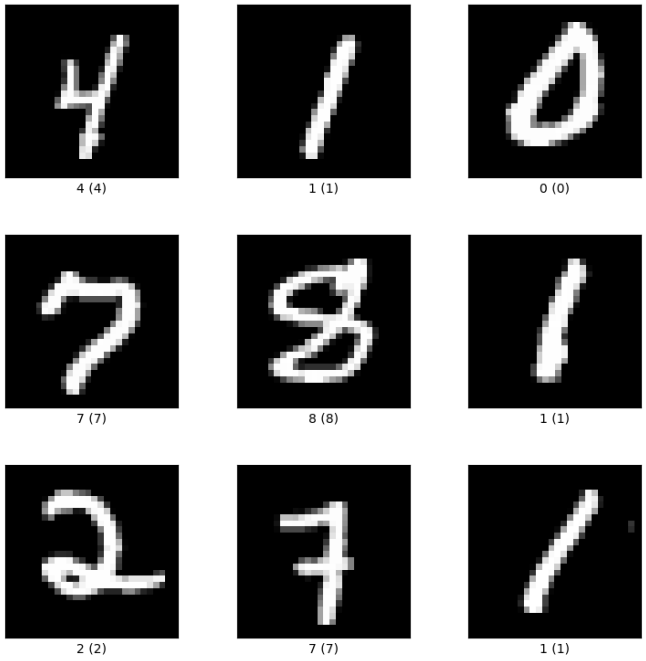


Figura 1.1: Exemplos de imagens do conjunto de dados MNIST.

SHRIKUMAR; GREENSIDE; KUNDAJE, 2017). Tomando como exemplo as redes neurais convolucionais, pode-se considerar como variáveis de entrada tanto os pixels de uma imagem quanto os mapas de características gerados pelas camadas intermediárias da rede. Nesse contexto, uma das técnicas de explicabilidade de redes neurais profundas é a DeepLIFT (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017), que funciona atribuindo valores de importância em função da diferença entre as variáveis de entrada e entradas de referência respectivas, sendo a referência escolhida de acordo com o problema em questão. Na prática a entrada de referência para uma entrada específica são dados que não possuem a propriedade que caracteriza a entrada em questão. Tomando como exemplo o conjunto de dados MNIST (LECUN et al., 1998) onde os dígitos são representados por pixels claros em um fundo preto - como pode ser visto na Figura 1.1 - uma possível entrada de referência é a matriz com todos os valores zerados, equivalente ao fundo preto das imagens do MNIST, pois não possui os pixels claros que caracterizam os dígitos presentes nas entradas, como pode ser visto na Figura 1.2.

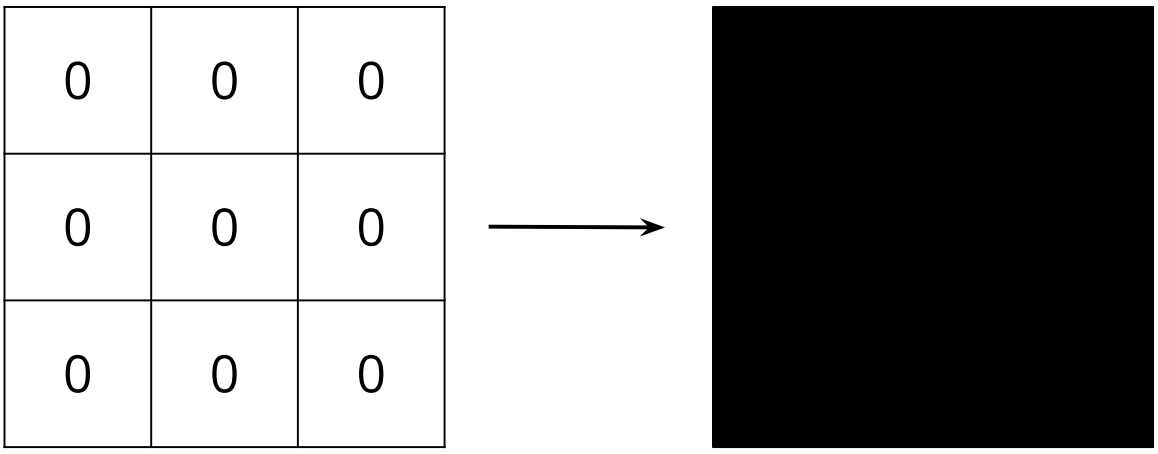


Figura 1.2: Exemplo de entrada de referência do método DeepLIFT quando usado no conjunto de dados MNIST. Para facilitar a visualização, a matriz de pixels foi simplificada para uma matriz 3×3, sendo que no conjunto de dados MNIST as matrizes são de 28×28 pixels.

Sabendo que as técnicas de explicabilidade associam valores de importância às características extraídas pela rede neural, esses valores de importância, em tese, podem ser usados como critério de seleção dos neurônios e/ou filtros a serem removidos durante o processo de poda. Os valores de importância atribuídos às características extraídas por um filtro convolucional, por exemplo, podem ser usadas como *proxy* da importância do próprio filtro convolucional, podendo assim substituir a magnitude dos pesos como critério de poda. Uma limitação óbvia da utilização de métodos de explicabilidade em comparação à utilização da magnitude dos pesos são os cálculos adicionais necessários para computação destas métricas.

1.2 Questões de Pesquisa

Levando em consideração as limitações identificadas e com o objetivo de avaliar o método proposto nesta pesquisa, as seguintes questões de pesquisa foram formuladas:

- QP1. A aplicação das técnicas de rebobinamento em redes neurais convolucionais podadas de forma estruturada possibilita o surgimento de bilhetes vencedores?

- QP2. Quais os trade-offs observados ao substituir o critério de poda do estado-da-prática (magnitude dos pesos) pelo método DeepLIFT?

1.3 Objetivos da Pesquisa

Tomando como base as questões de pesquisa apresentadas anteriormente, esta pesquisa tem por objetivo central contribuir com novas evidências empíricas que dão suporte à obtenção de bilhetes vencedores no contexto de poda estruturada em redes neurais convolucionais. Além disso, propõe-se a utilização do método de interpretabilidade DeepLIFT (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017) como uma alternativa à utilização da magnitude dos pesos como critério de seleção de elementos a serem podados.

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

- Avaliar experimentalmente o surgimento de bilhetes vencedores usando Rebobinamento dos Pesos (*Weight Rewinding*) em redes neurais convolucionais podadas de forma estruturada.

- Avaliar experimentalmente o surgimento de bilhetes vencedores usando Rebobinamento da Taxa de Aprendizagem (*Learning Rate Rewinding*) em redes neurais convolucionais podadas de forma estruturada.

- Comparar os resultados obtidos as técnicas de rebobinamento com as mesmas redes treinadas com pesos inicializados aleatoriamente.

- Avaliar o desempenho de redes neurais convolucionais podadas usando o método DeepLIFT como alternativa à magnitude dos pesos.

1.4 Contribuições

Esta dissertação contribui com a literatura de poda de redes neurais artificiais, apresentando novos experimentos que avaliam o surgimento de bilhetes vencedores quando aplicada a poda estruturada de redes neurais convolucionais. Além disso, vale destacar as seguintes contribuições:

- Comparação entre as abordagens de retreinamento associadas à Hipótese do Bilhete de Loteria.
- Análise do impacto das abordagens de poda global e local na identificação de bilhetes vencedores.
- Proposta de substituição da Magnitude dos Pesos pela métrica DeepLIFT como critério de seleção de filtros a serem podados em combinação com a abordagem de retreinamento de Rebobinamento da Taxa de Aprendizagem.
- Publicação do artigo intitulado "*Evaluating the Emergence of Winning Tickets by Structured Pruning of Convolutional Networks*" (MAGALHÃES et al., 2020) na *33rd Conference on Graphics, Patterns and Images* - SIBGRAPI 2020.

1.5 Organização do Trabalho

A estrutura deste documento está organizada como segue. No Capítulo 2, é apresentada a fundamentação teórica necessária para compreender o conteúdo do trabalho, como os conceitos relacionados às redes neurais, técnicas de compressão, poda de redes neurais convolucionais e sobre interpretabilidade de redes neurais. No Capítulo 3, são apresentados as pesquisas relacionadas à esta. No Capítulo 4, é apresentada a metodologia adotada no desenvolvimento da pesquisa. No Capítulo 5 são apresentados os resultados dos experimentos conduzidos e a discussão dos resultados obtidos. No Capítulo 6, são apresentadas as conclusões da pesquisa e as perspectivas para trabalhos futuros.

Capítulo 2

Fundamentação

Neste capítulo são apresentados os conceitos necessários para a compreensão desta pesquisa. Na Seção 2.1 são apresentados conceitos fundamentais sobre Redes Neurais Artificiais e sua instância de maior sucesso, as Redes Neurais Profundas. Na Seção 2.2 são descritos conceitos e técnicas relacionados à Poda de Redes Neurais.

2.1 Redes Neurais Artificiais

As Redes Neurais Artificiais (*Artificial Neural Networks*) se referem a modelos matemáticos inspirados na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência (JAIN; JIANCHANG MAO; MOHIUDDIN, 1996). Enquanto biologicamente os neurônios se comunicam com neurônios adjacentes através das sinapses, numa rede neural artificial as diversas unidades de processamento se conectam através de canais de comunicação associados a um valor de peso que na sua contraparte biológica representa a força da conexão sináptica. As Redes Neurais Profundas, por sua vez, são uma instância de Redes Neurais Artificiais e recebem esse nome principalmente por conta da grande quantidade de parâmetros e camadas ocultas que possuem.

As Redes Neurais Profundas, assim como outras técnicas de Aprendizado Profundo, podem realizar aprendizado de características. Isso significa que essas redes - a partir da junção de transformações simples porém não-lineares - possuem a capacidade de extrair características em variados níveis de abstração a partir dos dados de entrada em seu estado bruto, como por exemplo, os pixels de uma imagem, sem a necessidade de extrair as características

previamente. Essa junção de transformações não-lineares permite que modelos baseados em Aprendizado Profundo consigam representar funções altamente complexas (LECUN; BENGIO; HINTON, 2015).

As Redes Neurais Profundas podem ser categorizadas de acordo com o tipo de problema que resolvem, suas unidades funcionais e sua arquitetura, sendo estas as mais comuns:

- Redes Neurais Convolucionais (*Convolutional Neural Networks*);
- Redes Neurais Recorrentes (*Recurring Neural Networks*);
- *Long Short-Term Memory*;
- Autocodificadores (*Autoencoders*);
- Redes Adversárias Generativas (*Generative Adversarial Networks*).

Para desenvolvimento desta pesquisa, conduziram-se experimentos em Redes Neurais Convolucionais.

2.1.1 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (CNNs) são uma das categorias de Redes Neurais Profundas de maior sucesso, principalmente em tarefas de Visão Computacional dada a capacidade dessas redes em explorar correlação espacial nos dados.

As CNNs possuem uma arquitetura composta de múltiplas camadas e para cada uma dessas camadas tanto a entrada quanto a saída são mapas de características. Caso a entrada seja, por exemplo, uma imagem colorida, cada mapa de características será uma matriz contendo os valores de um canal de cor da imagem de entrada. Na saída, cada mapa de características representa uma característica particular extraída de todas as regiões da entrada. A arquitetura das CNNs costuma ser definida em estágios e cada estágio é composto de 3 tipos de camadas com funcionalidades distintas: Camada Convolucional, Camada de Ativação (ou de Não-Linearidade) e Camada de *Feature Pooling* (LECUN; KAVUKCUOGLU; FARABET, 2010).

As Camadas Convolucionais de uma CNN são compostas de filtros (ou *kernels*) convolucionais treináveis, onde cada um dos filtros é aplicado no mapa de características de

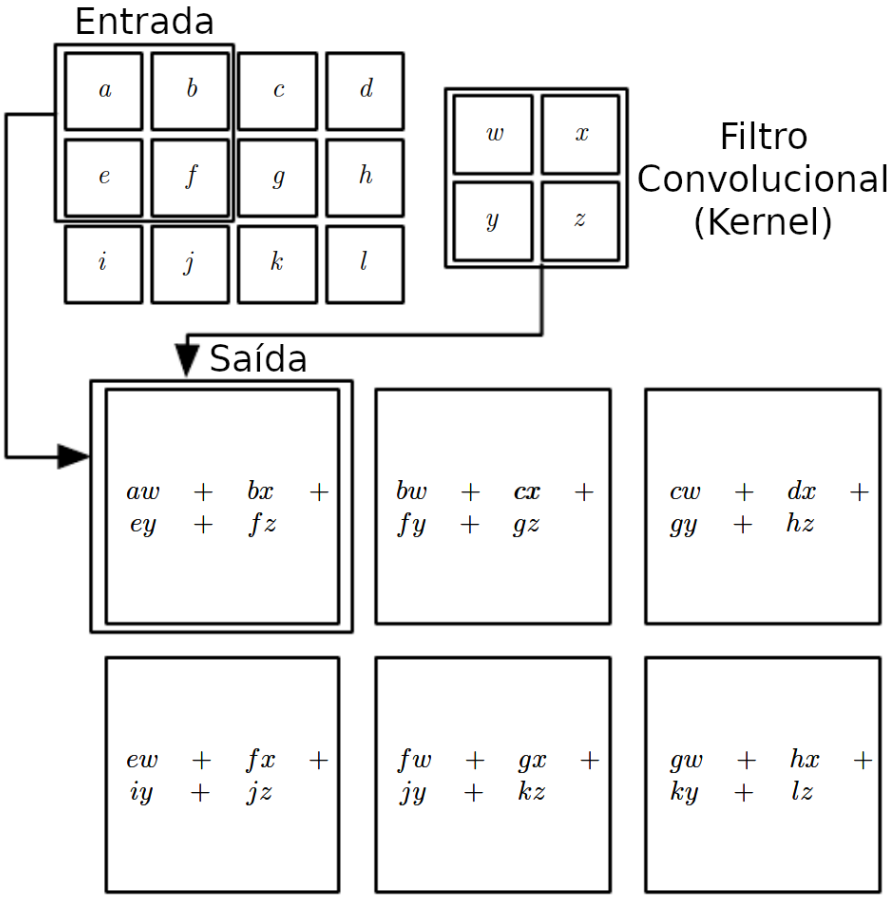


Figura 2.1: Exemplo de operação de Convolução. Cada caixa da saída é obtida ao deslizar a caixa da entrada e aplicar o filtro convolucional. Figura adaptada de Deep learning (2016).

entrada e então a saída é computada. Essa saída é obtida ao realizar uma operação de convolução discreta entre o mapa de características de entrada e o filtro convolucional, conforme ilustrado na Figura 2.1. As Camadas de Ativação em uma CNN são responsáveis por aplicar funções de ativação, não-lineares, nos mapas de características gerados pelas camadas convolucionais.

Por fim, as Camadas de *Feature Pooling* são responsáveis pela redução de dimensionalidade de todos os mapas de características gerados pelas camadas convolucionais e de ativação. O processo de *pooling* consiste em sumarizar os valores de diferentes regiões do mapa de características e as operações mais comumente adotadas em CNNs são *Average Pooling*, que consiste em calcular a média dos valores de uma região específica e *Max Pooling*, que consiste em considerar somente o maior dentre os valores de uma região. Ambas as operações de*pooling* são ilustradas na Figura 2.2. A redução de dimensão gerada pelas

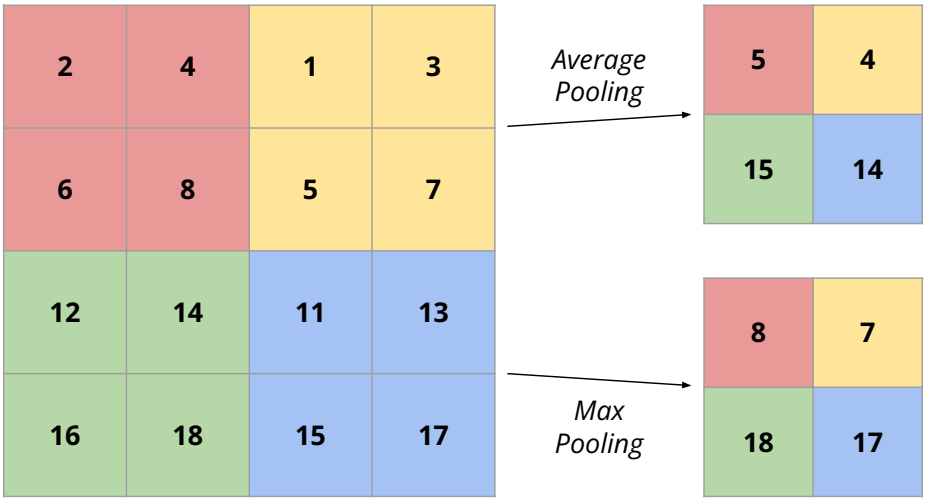


Figura 2.2: Exemplo de operações de *Average Pooling* e *Max Pooling*. As cores associam o conjunto de valores da entrada usados nas operações aos seus respectivos resultados.

camadas de *pooling* ajuda a fazer com que as representações extraídas se tornem invariantes a translações no mapa de características de entrada. Invariância a translação significa que se a entrada for transladada em uma pequena distância, os valores das saídas obtidos pelo processo de *pooling* não mudam (GOODFELLOW; BENGIO; COURVILLE, 2016).

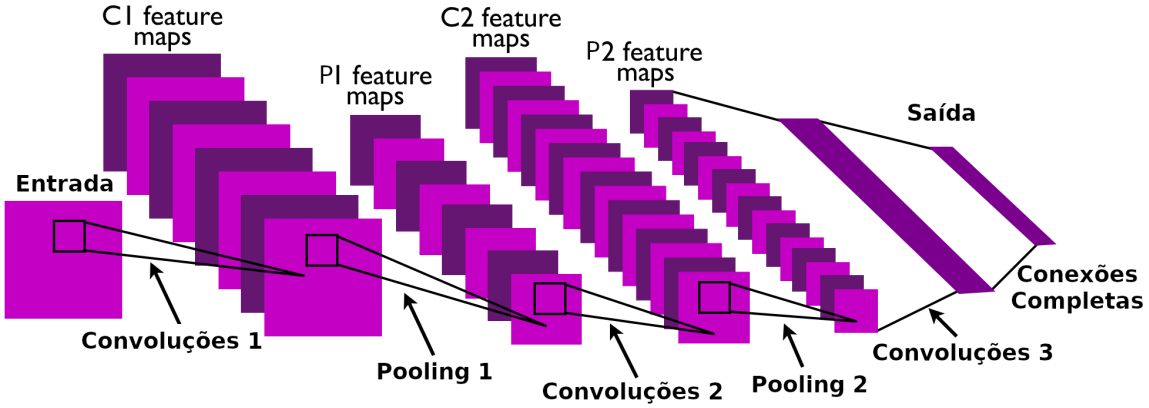


Figura 2.3: Arquitetura típica de uma Rede Neural Convolucional. Figura adaptada de Convolutional networks and applications in vision (2010).

Em tarefas de classificação, uma ou mais Camadas Completamente Conectadas (*Fully Connected Layers*) são adicionadas ao final da CNN com a finalidade de associar os atributos extraídos pelas camadas anteriores às diferentes classes a serem detectadas. Nesse contexto, funções de ativação específicas costumam ser aplicadas à saída da última camada

da CNN, sendo a função de ativação *Softmax* a mais comum para classificação multivariada e a função de ativação *Sigmoid* a mais comum para classificação binária (NWANKPA et al., 2018). A Figura 2.3 ilustra uma típica arquitetura de CNNs, mostrando os estágios e camadas constituintes mais comuns. A CNN usada no desenvolvimento desta pesquisa, a CNN VGG-16 (SIMONYAN; ZISSERMAN, 2015), é composta por estas mesmas camadas.

2.1.2 Avaliação de Redes Neurais Convolucionais

A avaliação de CNNs pode ser feita com base em medidas de desempenho e capacidade [1], permitindo assim a comparação de diferentes soluções e arquiteturas de CNNs. Em ambientes de produção, tais medidas funcionam como critérios e são fundamentais na hora de determinar a adoção ou não de uma solução baseada em CNN. Nesta pesquisa, adotou-se as seguintes medidas para avaliar as redes neurais convolucionais podadas: Acurácia e Número de Parâmetros.

Acurácia

A Acurácia é uma medida que indica quão bem um modelo está inferindo, considerando todo o conjunto de dados. Em um cenário de classificação multi-classe, considere a seleção de uma entrada aleatória i de um dado conjunto de dados $\mathcal{D}$ e a inferência da classe pelo modelo para esta entrada $f(i;\theta)$. Neste cenário a Acurácia representa a taxa de acerto do modelo para as diferentes entradas do conjunto de dados $\mathcal{D}$.

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

Em um cenário de classificação binária, a acurácia é calculada de acordo com a Fórmula 2.1, onde TP (*True Positives*) representa os casos positivos classificados corretamente, TN (*True Negatives*) representa os casos negativos classificados corretamente, FP (*False Positives*) representa os casos incorretamente classificados como positivos e FN (*False Negatives*) representa os casos incorretamente classificados como negativos.

[1] A capacidade de um modelo é a sua habilidade de se adaptar a uma grande variedade de funções. Uma forma de controlar a capacidade de um algoritmo de aprendizagem é definindo o conjunto de funções que o algoritmo de aprendizagem pode aproximar (GOODFELLOW; BENGIO; COURVILLE, 2016). Em Redes Neurais Profundas, a capacidade do modelo está relacionada à quantidade de parâmetros que ele possui.

$$\frac{T_{C1} + T_{C2} + T_{C3} + ... + T_{Cn}}{N} \tag{2.2}$$

No entanto, em classificação multi-classe, não há uma definição de *True Positives*, *True Negatives*, *False Positives* e *False Negatives*, uma vez que não há somente casos positivos e negativos e cada entrada pode ser classificada como pertencente a uma das n classes do conjuntos de dados. Em classificação multi-classe a acurácia é calcula de acordo com a Fórmula 2.2, onde T_{Cn} representa os casos corretamente classificados para a classe n e N representa o total de entradas do conjunto de dados.

No contexto específico de Poda de Redes Neurais Convolucionais, a medida de Acurácia é comumente usada para medir o impacto do método de poda no desempenho da rede podada, ao comparar-se a acurácia da rede podada com a acurácia do modelo original não podado.

Vale observar que, para o cálculo da acurácia, todas as entradas tem o mesmo peso e contribuem igualmente no valor de acurácia. Por conta disso, em casos onde há desbalanceamento dos dados, ou seja, a distribuição das entradas por classe de um conjunto de dados é desproporcional, a acurácia não é uma medida confiável pois tende a esconder erros de classificação em classes com menos entradas. Entretanto, os conjuntos de dados utilizados nesta pesquisa (CIFAR-10 e CIFAR-100 (KRIZHEVSKY, 2009)) são balanceados, com a mesma quantidade de entradas em cada uma de suas classes e portanto é possível usar a acurácia como medida de desempenho.

Número de Parâmetros

Como apresentado anteriormente na Seção 1.1, na medida em que as CNNs evoluíram, o número de parâmetros destas redes tiveram um grande aumento, passando de 60 mil parâmetros na rede LeNet-5 (LECUN et al., 1998) aos impressionantes 138 milhões de parâmetros da rede VGG-16 (SIMONYAN; ZISSERMAN, 2015). O número de parâmetros indica a capacidade da rede neural e impacta no tempo de execução da rede, uma vez que quanto maior o número de parâmetros maior o consumo de memória, armazenamento e maior o número de computações a serem realizadas tanto durante o treinamento, quanto para inferência.

Junto ao número de parâmetros, duas características das redes neurais ajudam a entender a forma como esses parâmetros estão distribuídos ao longo da arquitetura, que são eles: Profundidade e Largura. Em Redes Neurais Convolucionais, por exemplo, a Profundidade

indica a quantidade de camadas que a rede possui, enquanto a Largura indica a quantidade de filtros convolucionais e neurônios que a rede possui em suas camadas. Para algumas arquiteturas de CNNs, a quantidade de filtros convolucionais existentes em cada camada pode variar, variando deste modo a Largura. A poda estruturada, adotada no desenvolvimento desta pesquisa e detalhada na seção seguinte, reduz o número de parâmetros ao atuar na Largura das redes podadas, diminuindo a quantidade de filtros convolucionais por camada.

2.2 Poda de Redes Neurais

Como introduzido na Seção 1.1, Poda de Redes Neurais se refere a uma categoria bem estabelecida de técnicas de compressão de redes neurais que tem como objetivo reduzir o consumo de armazenamento, de memória e o uso de recursos computacionais, sem danos significativos à acurácia das redes podadas. Em resumo, as técnicas de poda de redes neurais atuam de modo a equilibrar as métricas apresentadas na subseção 2.1.2.

Uma vasta gama de técnicas de poda é estruturada de acordo com estas três etapas principais:

(1) Treinar a rede neural até a convergência.

(2) Podar a rede neural para que atinja um nível de esparsidade específico de acordo com alguma heurística.

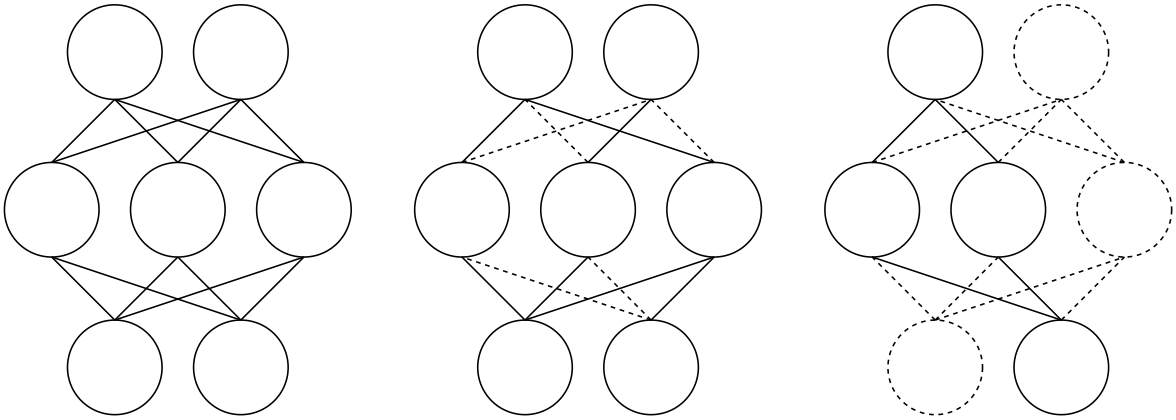
(3) Retreinar a rede podada para compensar qualquer eventual perda em acurácia e capacidade de generalização.

Entretanto, existem aspectos específicos que diferenciam as técnicas de poda e influenciam diretamente na rede neural que resulta do processo de poda. Alguns destes aspectos são descritos nas subseções a seguir.

2.2.1 Poda Não-Estruturada e Poda Estruturada

A poda não estruturada, também conhecida como poda de pesos, consiste na remoção de conexões entre neurônios e/ou filtros de camadas adjacentes. A remoção destas conexões é

efetuada introduzindo valores zero nas matrizes de pesos da rede neural, que em termos práticos equivale a remover o peso. Apesar da eficácia de métodos de poda não-estruturada, como a pesquisa de Frankle et al. (2019), sua aplicação ainda é limitada, uma vez que hardware de uso geral (ex. CPUs e GPUs) e bibliotecas computacionais (ex. BLAS) não lidam bem com matrizes esparsas, penalizando a execução de redes podadas desta forma (HAN et al., 2015).



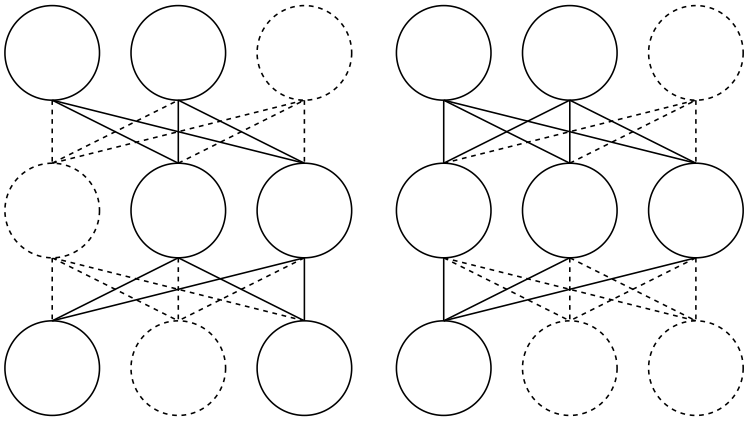
(a) Rede não-podada (b) Poda não-estruturada (c) Poda estruturada

Figura 2.4: Efeito das podas não-estruturada e estruturada sobre a arquitetura da rede neural. Unidades e conexões tracejadas indicam que elas foram podadas.

Já a poda estruturada, como o nome sugere, consiste na remoção de estruturas inteiras da rede neural (LI et al., 2017; HE, Y. et al., 2019). Tais estruturas a serem removidas podem ser neurônios de camadas completamente conectadas ou filtros convolucionais inteiros em camadas convolucionais. O principal benefício ao usar a poda estruturada é que, ao não acrescentar esparsidade à rede neural, a arquitetura resultante do processo de poda permite uma aceleração efetiva da rede em quase todas as bibliotecas de hardware e software. A Figura 2.4 mostra como a poda não estruturada e a poda estruturada afetam a arquitetura da rede neural.

2.2.2 Localidade da Poda

Algumas técnicas de poda, como nas pesquisas de Li et al. (2017) e de Yang He et al. (2019), exigem que uma taxa de poda definida pelo usuário seja aplicada ao modelo. Esta taxa de poda representa indiretamente o número de pesos, neurônios ou filtros a serem removidos durante o processo de poda. Independentemente da taxa de poda aplicada, é necessário defi-



(a) Rede podada localmente (b) Rede podada globalmente

Figura 2.5: Efeito das podas local e global sobre a arquitetura da rede neural. Objetos tracejados indicam que eles foram podados.

nir de onde, na rede a ser podada, os elementos serão removidos, seja local ou globalmente. A Figura 2.5 ilustra como a poda local e global afeta a arquitetura DNN.

Poda Local

Na poda local, os elementos são removidos de acordo com a sua localização na rede neural. Independente do critério adotado, os pesos, neurônios ou mesmo filtros convolucionais inteiros são ranqueados e podados de formal proporcional em todas as camadas da rede neural. Han et al. (2015) e Frankle e Carbin (2019) são exemplos de pesquisas que aplicam a poda local em alguns cenários.

Poda Global

Na poda global, adotada nas pesquisas de Frankle et al. (2019), Salama et al. (2019) e Renda, Frankle e Carbin (2020), os elementos são removidos sem levar em conta sua localização. Deste modo, para alcançar uma taxa de poda alvo, são aplicadas frações diferentes da taxa de poda em cada camada com base no ranqueamento global dos elementos da rede neural, mas garantindo que a quantidade de pesos, neurônios ou filtros removidos seja equivalente à taxa de poda alvo aplicada.

Vale notar que qualquer taxa de compressão contínua, utilizada para a poda, deve ser mapeada em um número discreto representando o número de elementos a serem efetivamente

removidos. Assim, o número de elementos a serem removidos nem sempre corresponderá à taxa de compressão escolhida, mas, ao invés disso, a mais próxima possível. Além disso, a poda global pode inviabilizar uma rede neural, dependendo do nível de compressão aplicado e do critério de seleção dos elementos a serem removidos. Este problema pode acontecer, por exemplo, quando todos os pesos, neurônios ou filtros são completamente removidos de uma determinada camada. Além disso, não é permitido remover elementos da camada de saída, pois ela é responsável pelas saídas da rede neural.

2.2.3 Temporalidade (*Scheduling*) da Poda

Uma vez definida a localidade da poda, é preciso definir também a temporalidade da poda. A temporalidade pode ser amplamente distribuída em três categorias (LIU, Zhuang et al., 2019; RENDA; FRANKLE; CARBIN, 2020), como se segue:

- **Poda *One-Shot*.** A poda é aplicada à rede neural treinada de uma só vez, de modo que a taxa de poda alvo seja alcançada ao final do processo.
- **Poda Iterativa.** Nesta abordagem, pequenas frações da rede neural são podadas ao final de um ciclo de treinamento. Assim, a taxa de poda alvo é alcançada ao final de várias iterações de ciclos de treinamento e processos de poda.
- **Poda Gradual.** Nesta abordagem, a rede neural é podada durante todo o processo de treinamento, para que ao final do ciclo de treinamento seja produzida uma rede neural podada.

Devido à integração entre o ciclo de treinamento e o processo de poda na Poda Gradual, não há necessidade de uma etapa adicional de retreinamento da rede podada, já que a poda é feita em conjunto com o treinamento da rede neural. Entretanto, na Poda *One-Shot* e na Poda Iterativa, cujos elementos são removidos somente ao final do processo de treinamento, as redes podadas muitas vezes precisam ser retreinadas para compensar a perda de informação ao ser podada.

Além disso, nesta pesquisa avaliamos somente redes podadas obtidas por Poda Iterativa, uma vez que pesquisas anteriores indicam que a poda iterativa é capaz de gerar redes podadas com maior nível de compressão e menor perda de acurácia (HAN et al., 2015; RENDA;

FRANKLE; CARBIN, 2020; MORCOS et al., 2019; FRANKLE; CARBIN, 2019) quando comparadas às redes obtidas por poda *one-shot* ou poda gradual.

2.2.4 Poda Baseada na Magnitude dos Pesos

O critério de poda baseado na magnitude dos pesos é intuitivo e surpreendentemente eficiente, sendo utilizado em pesquisas recentes e relevantes (GALE; ELSEN; HOOKER, 2019; HAN et al., 2015; MORCOS et al., 2019). Esse critério consiste na remoção dos pesos que apresentam menor magnitude (valor absoluto) e como já mencionado anteriormente na Seção 1.1, a intuição que leva à utilização da magnitude dos pesos como critério de poda é que pesos que possuem valores baixos de magnitude produzem valores baixos de ativação e portanto impactam menos na saída da rede neural (HE, Y. et al., 2018).

Quando a poda baseada na magnitude dos pesos é aplicada em poda não-estruturada, os pesos são removidos individualmente com base nos valores magnitude. Quando aplicada em poda estruturada de CNNs, uma vez que o objetivo é remover filtros convolucionais inteiros, as normas (mais comumente L1 ou L2) das matrizes de pesos dos filtros convolucionais são calculadas e os filtros que apresentarem menores valores de norma são podados.

2.2.5 Métodos de Explicabilidade e Poda de Redes Neurais

Devido ao enorme sucesso em diversas tarefas, como apresentado no Capítulo 1, o uso de soluções baseadas em redes neurais profundas aumentou significativamente nos últimos anos, em especial as CNNs. Entretanto, uma característica dessas redes as impedem de serem utilizadas em tarefas críticas: as redes neurais profundas são soluções caixa-preta, isto é, a estrutura do modelo não dá indícios sobre a função que ele modela, sendo o oposto de modelos interpretáveis como equações de regressão e árvores de decisão, por exemplo. Entretanto, pesquisas recentes se dedicaram em entender o funcionamento das redes neurais profundas e torná-las mais interpretáveis, aumentando a confiança dos usuários. Tais pesquisas constituem o campo que é chamado de Inteligência Artificial Explicável (*Explainable Artificial Intelligence* - XAI).

No contexto de CNNs, a abordagem mais comum de explicabilidade são os métodos de saliência, métodos de interpretabilidade visual que se baseiam principalmente na represen-

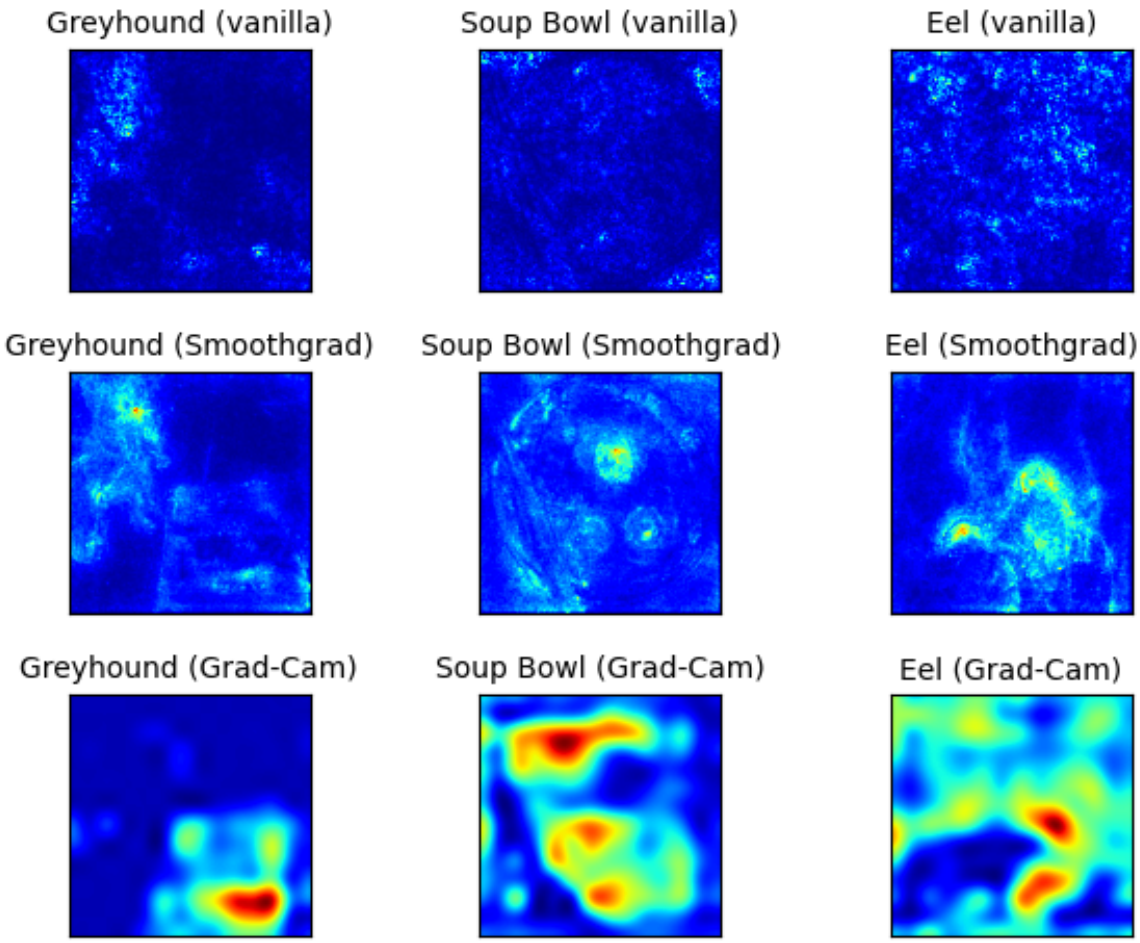


Figura 2.6: Mapas de Saliência dos métodos *Image-Specific Class Saliency - Vanilla* (SIMONYAN; VEDALDI; ZISSERMAN, 2013), *Smoothgrad* (SMILKOV et al., 2017) e *Grad-Cam* (SELVARAJU et al., 2016) para três diferentes entradas. Fonte: Interpretable Machine Learning (2019).

tação das pontuações de importância para uma determinada entrada na forma de mapas de saliência *(saliency maps)*, ou seja, para cada pixel da imagem de entrada, ou cada característica do mapa de características gerado pelas camadas convolucionais, um valor de importância é atribuído. Esses mapas de saliência são usualmente representados como mapas de calor (*heatmap*) que destacam as regiões mais relevantes da entrada ou mapa de características, como pode ser observado na Figura 2.6. *Conductance* (DHAMDHERE; SUNDARARAJAN; YAN, 2019), DeepLIFT (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017) e *Layer-Wise Relevance Propagation* (MONTAVON, Grégoire et al., 2019) são exemplos dessa categoria de métodos. Essa capacidade de atribuir valores de importância para neurô-

nios, filtros ou até mesmo camadas de uma rede neural convolucional é o que possibilita a adoção desses métodos como parte do processo de poda, uma vez que os valores de importância obtidos por esses métodos podem ser usados como critério de seleção dos elementos a serem removidos durante a poda.

Em Shrikumar, Greenside e Kundaje (2017), os autores apresentam o método DeepLIFT, que atribui pontuações de importância ao comparar a ativação de cada neurônio para uma determinada entrada com a ativação na entrada de referência. No caso de classificação de imagens com CNNs, a entrada de referência é uma imagem que não possui as característica que melhor descrevem a imagem de entrada. Os autores definem formalmente o método DeepLIFT como segue. Seja t um neurônio de saída de interesse e sejam $x_1, x_2, ..., x_n$ alguns dos neurônios das camadas intermediárias necessários e suficientes para o cálculo de t. Seja t^0 a ativação na entrada de referência de t. A medida Δt é a diferença-da-referência, que é $\Delta t = t - t^0$. DeepLIFT atribui pontuações de contribuição $C_{\Delta x_i \Delta t}$ à Δx_i tal que:

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta t} = \Delta t \tag{2.3}$$

$C_{\Delta x_i \Delta t}$ pode ser interpretado como a quantidade de diferença-da-referência em t que é proveniente da diferença-da-referência em x_i.

Nesta pesquisa optou-se por utilizar o método DeepLIFT por duas razões principais:

1. O método DeepLIFT resolve duas limitações dos métodos baseados em gradiente: (i) É capaz de atribuir valores de importância mesmo em situações onde o gradiente é zero. (ii) A variação nos valores de importância é suave e contínua, enquanto que nos métodos baseados em gradiente, a natureza descontínua dos gradientes faz com que aconteçam variações súbitas nos valores de importância mesmo com alterações mínimas na entrada.

2. O método DeepLIFT é mais computacionalmente eficiente que os métodos baseados em perturbação, que requerem uma propagação na rede para cada alteração na imagem de entrada.

Entretanto, o método DeepLIFT possui a limitação de que os valores de importância só são calculáveis quando se é definida a entrada de referência. Em Shrikumar, Greenside e

Kundaje (2017), os autores sugerem as seguintes possibilidade de entradas de referência: (i) uma imagem nula com todos os *pixels* zerados, (ii) uma imagem que é a média das imagens na partição de treinamento do conjunto de dados usado, e (iii) uma versão borrada da imagem de entrada.

Capítulo 3

Pesquisas Relacionadas

Neste capítulo são apresentadas e discutidas as pesquisas relacionadas. Esta pesquisa é relacionada principalmente com a literatura de poda de redes neurais, mais especificamente poda estruturada e a Hipótese do Bilhete de Loteria. Ademais, esta pesquisa também se relaciona com a literatura de interpretabilidade de redes neurais. Este capítulo está organizado de modo que as referências são listadas e contextualizadas de acordo com a ordem de publicação.

As pesquisas intituladas *Optimal brain damage* (LECUN; DENKER; SOLLA, 1990) e *Second order derivatives for network pruning: Optimal brain surgeon* (HASSIBI; STORK, 1993) são pioneiras na literatura de poda de redes neurais e se baseiam na ideia de estimar *a priori* o impacto da remoção de conexões e/ou neurônios das redes a serem podadas. Em ambas as pesquisas é usada uma métrica de saliência (*saliency*), calculada como uma série de Taylor[1], definida como a estimativa do aumento no erro do treinamento caso um dado peso w seja removido e os demais pesos atualizados. Uma vez que remover individualmente cada peso da rede e medir o impacto no erro do treinamento seria proibitivamente custoso, os autores constroem um modelo local da função de erro e predizem com base nesse modelo a saliência, aproximando a função de erro à uma série de Taylor fazendo uso de matrizes hessianas[2]. Ainda assim, é preciso simplificar o problema, uma vez que a matriz *hessiana*

[1] A série de Taylor de uma função é uma soma infinita dos termos que são expressos como termos das derivadas dessa função em um único ponto.

[2] A matriz Hessiana H de uma função f de n variáveis é a matriz quadrada com n colunas e n linhas das derivadas parciais de segunda ordem da função.

dos parâmetros de uma rede neural é enorme, mesmo para redes consideradas simples para os padrões atuais. Tomando como exemplo a rede usada em LeCun, Denker e Solla (1990), a matriz *hessiana* dos parâmetros de uma rede com 2600 parâmetros teria $6,5 \times 10^6$ termos, o que implica em custo computacional elevado para computar todos os termos. Para contornar essa limitação, os autores assumem que a saliência causada pela remoção de vários elementos da rede neural é igual a soma da saliência causada pela remoção de cada elemento individual, simplificando o problema ao cálculo da diagonal da matriz *hessiana*, ou seja, ao cálculo das derivadas de segunda ordem que compõem a diagonal. Entretanto, apesar dessa simplificação viabilizar a execução desses métodos em redes com poucos parâmetros, em redes superparametrizadas (e.g. VGG (SIMONYAN; ZISSERMAN, 2015) e ResNet (HE, K. et al., 2016)) o cálculo dessas derivadas é computacionalmente custoso, penalizando a adoção desses métodos. Entretanto, apesar dessas limitações, abordagens derivadas de LeCun, Denker e Solla (1990) e Hassibi e Stork (1993) continuam sendo investigadas em pesquisas recentes como, por exemplo, Peng et al. (2019), que propõe um novo algoritmo que reduz o custo de computação da matriz Hessiana ao levar em consideração a relação entre os filtros/canais da rede e Huibo Wu et al. (2021), que adapta o algoritmo de poda apresentado em Hassibi e Stork (1993) para o contexto de Pré-distorção Digital (Digital Predistortion), com a finalidade de reduzir a complexidade do algoritmo, removendo coeficientes redundantes.

Como alternativa aos métodos baseados em séries de Taylor, têm-se os métodos baseados na magnitude dos pesos. Esses métodos partem da premissa de que a magnitude é um indicativo de importância dos pesos e portanto pode ser usada como critério para seleção de conexões e/ou neurônios a serem removidos em uma rede neural. Uma vez que o cálculo da magnitude dos pesos é mais simples e menos custoso computacionalmente que o cálculo de derivadas de segunda ordem, tais métodos são escaláveis e podem ser aplicados em redes neurais profundas superparametrizadas. Apesar da validade da relação entre magnitude dos pesos e importância dos pesos ser questionada desde as pesquisas seminais da literatura de poda de redes neurais (LECUN; DENKER; SOLLA, 1990; HASSIBI; STORK, 1993), pesquisas recentes têm defendido a efetividade da aplicação de abordagens baseadas na magnitude dos pesos (HAN et al., 2015; LI et al., 2017; FRANKLE; CARBIN, 2019; HE, Y. et al., 2019; ZHOU, H. et al., 2019), sendo a pesquisa de Frankle e Carbin (2019)

considerada uma das pesquisas recentes mais influentes na literatura recente de Poda de Redes Neurais, na qual a Hipótese do Bilhete de Loteria (*The Lottery Ticket Hypothesis*) foi proposta. Maiores detalhes são apresentados na próxima seção.

3.1 Pesquisas Relacionadas à Hipótese do Bilhete de Loteria

Na Hipótese do Bilhete de Loteria, proposta por Frankle e Carbin (2019), formula-se que é possível encontrar sub-redes esparsas (redes podadas), a partir de uma rede neural superparametrizada, e treiná-las partindo de pesos aleatórios iniciais para que as sub-redes esparsas possam atingir acurácia semelhante ou até mesmo maior do que a sua contraparte não podada.

A seguir tem-se a definição formal da Hipótese do Bilhete de Loteria, extraída de Frankle e Carbin (2019): Considere uma rede neural *feed-forward* densa $f(x;\theta)$ com parâmetros iniciais $\theta = \theta_0 \sim \mathcal{D}_\theta$. Quando otimizada com gradiente descendente estocástica (*Stochastic Gradient Descent - SGD*) em um conjunto de treinamento arbitrário, f alcança a perda mínima no conjunto de validação l na iteração j com acurácia no conjunto de teste a. Além disso, considere treinar $f(x; m \odot \theta)$ com uma máscara de poda[3] $m \in \{0,1\}^{|\theta|}$ aplicada aos seus parâmetros de modo que sua inicialização é $m \odot \theta_0$. Quando otimizada com SGD no mesmo conjunto de treinamento (com m fixado), f alcança a perda mínima no conjunto de validação l' na iteração j' com acurácia no conjunto de teste a'. A hipótese do bilhete de loteria prediz que $\exists m$ para o qual $j' \leq j$ (tempo de treinamento proporcional), $a' \geq a$ (acurácia proporcional), e $||m||_1 \ll |\theta|$ (menos parâmetros).

Tais sub-redes, $f(x; m \odot \theta)$, recebem o nome de bilhetes vencedores e são obtidas a partir das seguintes etapas:

(1) Inicialize aleatoriamente a rede neural a ser podada.

(2) Treine-a até que convirja.

[3]Uma máscara de poda é uma matriz binária que específica quando um peso ou conjunto de pesos deve ser podado (0) ou não (1). A rede podada ($f(x; w \odot m)$) resulta do produto Hadamard ($\odot$) entre a matriz de pesos (w) e da máscara de poda (m).

(3) Pode uma parcela da rede usando a magnitude dos pesos como critério de seleção das conexões e/ou neurônios a serem podados.

(4) Redefina os pesos restantes da rede podada para seus respectivos valores aleatórios iniciais.

(5) Treine a rede podada até que convirja.

Os autores referem-se à aplicação dessas etapas de forma iterativa por Poda Iterativa Baseada em Magnitude (*Iterative Magnitude Pruning - IMP*). Com a aplicação da IMP, os autores foram capazes de obter redes podadas em níveis agressivos (menores que 10-20% do tamanho da rede original não podada), as quais alcançam ou até mesmo superam a acurácia de suas contrapartes não podadas. Além disso, as redes podadas convergiram em menos épocas de treinamento, quando comparadas às redes originais não podadas.

Apesar dos excelentes resultados obtidos em redes totalmente conectadas e redes convolucionais rasas, esta abordagem não foi capaz de encontrar bilhetes vencedores em redes neurais convolucionais profundas. Com o objetivo de superar essa limitação, os autores propuseram, em Frankle et al. (2019), uma versão atualizada da Hipótese do Bilhete de Loteria, modificando os passos necessários para encontrar bilhetes vencedores. Em vez de redefinir os pesos para seus valores aleatórios iniciais W_0, os pesos W são agora redefinidos para os valores que tinham em uma das épocas iniciais do treinamento da rede original não podada, W_k, para $k \ll T$, sendo T o número de iterações do treinamento da rede original não podada. A definição formal, extraída de Frankle et al. (2019), é como segue: Considere uma rede neural densa inicializada aleatoriamente $f(x; W_0)$ que é treinada para a acurácia a^* em T^* iterações. Seja W_t os pesos na iteração t do treinamento. Existe uma iteração $k \ll T^*$ e uma máscara de poda fixa $m \in \{0,1\}^{|W_0|}$ (onde $||m||_1 \ll |W_0|$) de tal modo que a sub-rede $f(x; m \odot W_k)$ é treinada para a acurácia $a \geq a^*$ em $T \leq T^* - k$ iterações. Com essa abordagem, denominada Hipótese do Bilhete de Loteria com Rebobinamento, os autores foram capazes de encontrar bilhetes vencedores em redes neurais convolucionais profundas.

Devido à efetividade da Hipótese do Bilhete de Loteria, pesquisas subsequentes concentraram-se nos aspectos teóricos dos bilhetes vencedores. Para entender quais aspectos da Hipótese do Bilhete de Loteria são determinantes para seu sucesso, Hattie Zhou et al. (2019) avaliaram empiricamente as máscaras de poda que emergem com os bilhetes vence-

dores. Para tal, os autores experimentaram com diferentes critérios de definição da máscara de poda (critério de poda), como por exemplo, manter os menores pesos treinados, manter os menores pesos obtidos na inicialização, manter os maiores pesos obtidos na inicialização, ou até mesmo manter os pesos que tiveram maior variação na magnitude ao longo do treinamento. Além disso, os autores também avaliaram a importância do rebobinamento dos pesos e do zeramento dos pesos durante o processo de poda para o sucesso dos bilhetes vencedores. Com base nesses experimentos, as principais conclusões de Hattie Zhou et al. (2019) são que:

- O critério de poda que mantém os pesos que tiveram maior variação na magnitude ao longo do treinamento desempenha tão bem quanto o que mantém os pesos com maior magnitude;
- O zeramento dos pesos é crucial para o sucesso dos bilhetes vencedores.

Além disso, os autores descobriram a existência de super-máscaras. Super-máscaras são máscaras de poda que ao serem aplicadas para podar redes não treinadas fazem com que essas redes podadas desempenhem muito melhor que redes podadas aleatoriamente ou inicializadas aleatoriamente.

Na pesquisa de Morcos et al. (2019), os autores focaram em avaliar o potencial de generalização dos bilhetes vencedores, isto é, quão bem um bilhete vencedor identificado usando um conjunto de dados específico e um otimizador específico desempenha se reutilizado no retreinamento em um conjunto de dados diferente ou com um otimizador diferente. Para avaliar a generalização entre conjuntos de dados, diferentes conjuntos de dados (CIFAR-10/CIFAR-100(KRIZHEVSKY, 2009), ImageNet (RUSSAKOVSKY et al., 2015), Places365 (ZHOU, B. et al., 2017)) foram utilizados para identificação dos bilhetes vencedores e em seguida, os bilhetes vencedores foram utilizados para treinamento em um conjunto de dados diferente do original. Para avaliar a generalização entre otimizadores, os otimizadores Adam (KINGMA; BA, 2014) e SGD foram utilizados para identificação dos bilhetes vencedores e em seguida, os bilhetes vencedores foram utilizados para treinamento com um otimizador diferente do original. Com base nestes experimentos, os autores concluem que:

- Bilhetes vencedores identificados em conjuntos de dados menores, e.g. CIFAR-10 e CIFAR-100, desempenham bem quando usados no treinamento em um conjunto

de dados maior, e.g. ImageNet, mas desempenham pior que os bilhetes vencedores identificados diretamente no conjunto de dados maior.

- Bilhetes vencedores identificados em conjuntos de dados maiores utilizados no treinamento em um conjunto de dados menor desempenham ainda melhor que os bilhetes vencedores identificados diretamente no conjunto de dados menor.

- Bilhetes vencedores identificados usando Adam pode ser utilizado no treinamento usando SGD e vice-versa, desde que as taxas de aprendizagem sejam devidamente ajustadas.

Em uma pesquisa mais recente, Renda, Frankle e Carbin (2020) introduziram o método de Rebobinamento da Taxa de Aprendizagem, uma nova abordagem de retreinamento alternativa ao Rebobinamento dos Pesos. Ao contrário do rebobinamento dos pesos, nessa abordagem os pesos finais obtidos no treinamento da rede não podada são mantidos (de forma similar ao método de *fine-tuning*), sendo redefinido apenas o protocolo de variação da taxa de aprendizagem ao retreinar a rede podada. Para avaliar a efetividade do Rebobinamento da Taxa de Aprendizagem, os autores comparam os resultados de três diferentes abordagens de retreinamento: *Fine-Tuning*, Reboninamento dos Pesos e Rebobinamento da Taxa de Aprendizagem. Os experimentos são conduzidos nas redes ResNet-56 (HE, K. et al., 2016) treinada no conjunto de dados CIFAR-10, ResNet-34 e ResNet-50 (HE, K. et al., 2016) treinada no conjunto de dados ImageNet e por fim, Google Neural Machine Translation (GNMT) (WU, Y. et al., 2016) treinada no conjunto de dados WMT'16 EN-DE. As abordagens de retreinamentos são avaliados de acordo com três critérios: Acurácia (A acurácia da rede podada resultante), Eficiência (Os recursos necessários para armazenar e executar a rede podada) e Custo de Busca (O custo para obter a rede podada).

Com base nos resultados dos experimentos, os autores concluem que:

- Os métodos de Rebobinamento superam a abordagem de *fine-tuning*. Tanto para poda estruturada, quanto não estruturada.

- O Rebobinamento da Taxa de Aprendizagem supera o Rebobinamento dos Pesos. Além disso, ao contrário do Rebobinamento dos Pesos que pode falhar quando rebobinado para os pesos iniciais aleatórios W_0, o Rebobinamento da Taxa de Apren-

dizagem na maioria dos casos se beneficia do rebobinamento do protocolo da taxa de aprendizagem para o início do treinamento.

3.2 Pesquisas em Poda Baseada em Explicabilidade

Métodos de explicabilidade de redes neurais podem ser utilizados como uma alternativa à magnitude dos pesos como critério de seleção de neurônios e/ou filtros a serem removidos durante o processo de poda, como já visto na subseção 2.2.5. Em Yeom et al. (2021), os autores usam o método de explicabilidade *Layer-Wise Relevance Propagation* (LRP) (MONTAVON, Grégoire et al., 2019) como critério de seleção de poda. Mais especificamente, os experimentos são focados em poda estruturada e transferência de aprendizado, onde os parâmetros de uma rede pré-treinada em um domínio de origem são ajustados (*Fine-Tuning*) em um domínio alvo. Com base nos experimentos, os autores apontam que a adoção do critério LRP demonstrou desempenho superior à magnitude dos pesos e demais métodos comparados, em cenários com e sem retreinamento.

A pesquisa de Sabih, Hannig e Teich (2020) propõe a utilização do método de explicabilidade DeepLIFT (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017) nos contextos de poda não-estruturada, estruturada e quantização. Com base nos resultados, os autores apontam que com a adoção do método DeepLIFT foram capazes de obter resultados do estado-da-arte ou competitivos tanto no contexto de poda quanto no contexto de quantização.

3.3 Considerações Finais

Na Tabela 3.1 é apresentado um resumo das abordagens das pesquisas relacionadas descritas na Seção 3.1 e da abordagem apresentada nesta pesquisa. As dimensões utilizadas para construção da tabela são o tipo de poda (estruturada ou não-estruturada), critério de poda, localidade da poda e abordagem de retreinamento. Essas dimensões foram apresentadas em detalhes na Seção 2.2.

Com base nas pesquisas relacionadas à Hipótese do Bilhete de Loteria apresentadas neste capítulo, observou-se que a maior parte dos achados e conclusões apresentados se baseiam em experimentos com poda não estruturada, sendo a pesquisa de Renda, Frankle e Carbin

(2020) a única exceção, apresentando resultados preliminares com poda estruturada. No entanto os experimentos conduzidos em poda estruturada se limitam à utilização de taxas de poda por camada minuciosamente selecionadas por Li et al. (2017) e um conjunto de taxas de poda derivado dessas taxas minuciosamente selecionadas.

Além disso, observou-se que nenhuma das pesquisas anteriores focou em avaliar o surgimento de bilhetes vencedores ao aplicar os métodos de Rebobinamento dos Pesos e Rebobinamento da Taxa de Aprendizagem para a realização de poda global estruturada, onde a arquitetura resultante do processo de poda é determinada automaticamente durante a execução da poda. Além disso, das pesquisas relacionadas à Hipótese do Bilhete de Loteria, somente a pesquisa de Hattie Zhou et al. (2019) experimenta com diferentes critérios de poda, embora todos derivados da magnitude dos pesos, enquanto nesta pesquisa propõe-se a utilização do método de interpretabilidade DeepLIFT (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017) como uma alternativa à utilização da magnitude dos pesos na abordagem de rebobinamento da taxa de aprendizagem.

Apesar das pesquisas descritas na Seção 3.2 apresentarem abordagens similares à utilização do método DeepLIFT proposta nesta dissertação, nenhuma delas foca na identificação de bilhetes vencedores e não fazem uso dos métodos de retreinamento relacionados à Hipótese do Bilhete de Loteria, como o método de Rebobinamento da Taxa de Aprendizagem, e se limitam ao *Fine-Tuning* das redes podadas. Além disso, quando da implementação e realização dos experimentos desta pesquisa, nenhuma das pesquisas descritas na Seção 3.2 haviam sido publicadas.

Por fim, vale ressaltar que neste capítulo focou-se em apresentar e discutir as pesquisas diretamente relacionadas às abordagens de poda utilizadas no desenvolvimento desta pequisa: Poda Basaeda na Magnitude dos Pesos, A Hipótese do Bilhete de Loteria e Poda Baseada em Explicabilidade. Por conta disso, outras abordagens de poda de redes neurais profundas não foram incluídas, como por exemplo: Poda Baseada em Meta-Aprendizado (LIU, Zechun et al., 2019; ZHANG et al., 2021) e Poda Baseada em Aprendizagem por Reforço (CHEN; CHEN; PAN, 2020; WANG; LI, 2022).

No próximo capítulo são apresentados detalhes sobre a metodologia utilizada nesta pesquisa.

Pesquisa	**Poda**	**Critério**	**Localidade**	**Retreinamento**
LeCun, Denker e Solla (1990)	Não-Estruturada	Série de Taylor	Global	Não especificado
Hassibi e Stork (1993)	Não-Estruturada	Série de Taylor	Global	Não especificado
Frankle e Carbin (2019)	Não-Estruturada	Magnitude	Global	Rebobinamento dos Pesos (W_0)
Frankle et al. (2019)	Não-Estruturada	Magnitude	Global	Rebobinamento dos Pesos ($W_t \mid t \geq 0$)
Hattie Zhou et al. (2019)	Não-Estruturada	Magnitude (várias variações)	Global	"
Morcos et al. (2019)	Não-Estruturada	Magnitude	Global	"
Renda, Frankle e Carbin (2020)	Estruturada e Não-Estruturada	Magnitude	Global	", Rebobinamento da Taxa de Aprendizagem e *Fine-Tuning*
Yeom et al. (2021)	Estruturada	*Layer-Wise Relevance Propagation*	Global	*Fine-Tuning*
Sabih, Hannig e Teich (2020)	Estruturada e Não-Estruturada	DeepLIFT	Global e Local	*Fine-Tuning*
Esta Pesquisa	Estruturada	Magnitude e DeepLIFT	Global e Local	Rebobinamento dos Pesos e Rebobinamento da Taxa de Aprendizagem

Tabela 3.1: Resumo das Pesquisas Relacionadas.

Capítulo 4

Materiais e Métodos

Neste capítulo são apresentados os materiais e métodos associados a esta pesquisa. Nesse sentido, são descritos a arquitetura da rede neural convolucional, os conjuntos de dados e as abordagens de poda adotadas nos experimentos conduzidos. Os experimentos especificados na Seção 4.2 têm como objetivo responder às questões de pesquisas apresentadas na Seção 1.2.

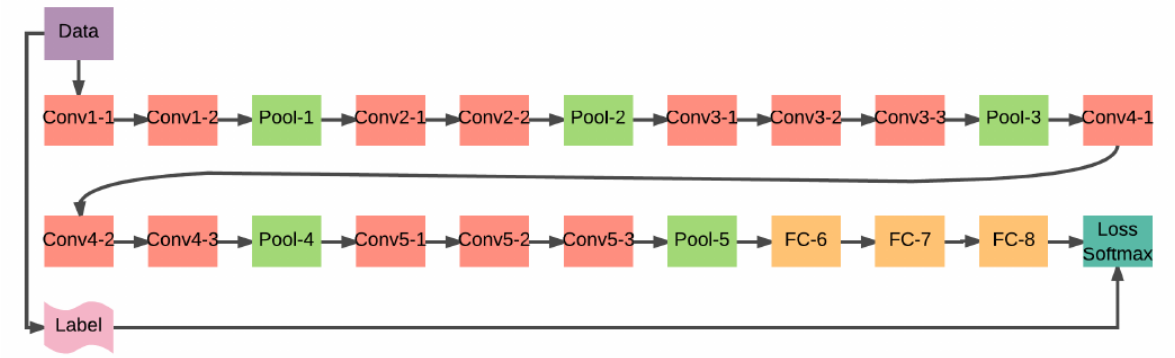


Figura 4.1: Arquitetura da rede neural convolucional VGG-16. Fonte: Qassim, Verma e Feinzimer (2018).

4.1 Materiais

Para realização dos experimentos, utilizou-se o modelo de rede neural convolucional VGG16 (SIMONYAN; ZISSERMAN, 2015), com funções de ativação *ReLU* (NAIR; HINTON, 2010) e camadas de *Batch Normalization* (IOFFE; SZEGEDY, 2015). Essa arquitetura

foi projetada inicialmente para a tarefa de classificação do desafio ILSVRC-2014 (RUSSAKOVSKY et al., 2015) e é bastante conhecida por sua capacidade, uma vez que é uma rede bastante profunda e com uma quantidade elevada de camadas completamente conectadas. Apesar de eficiente na classificação, alcançando 92,7% de acurácia no top-5 no teste do conjunto de dados ImageNet (RUSSAKOVSKY et al., 2015), a capacidade da arquitetura implica em um modelo computacionalmente custoso, grande em termos de armazenamento e consumo de memória e com tempo de treinamento elevado (QASSIM; VERMA; FEINZIMER, 2018). O custo computacional elevado e a facilidade de modificar a arquitetura são características que fazem do VGG16 um modelo atrativo para avaliação de técnicas de poda de redes neurais.

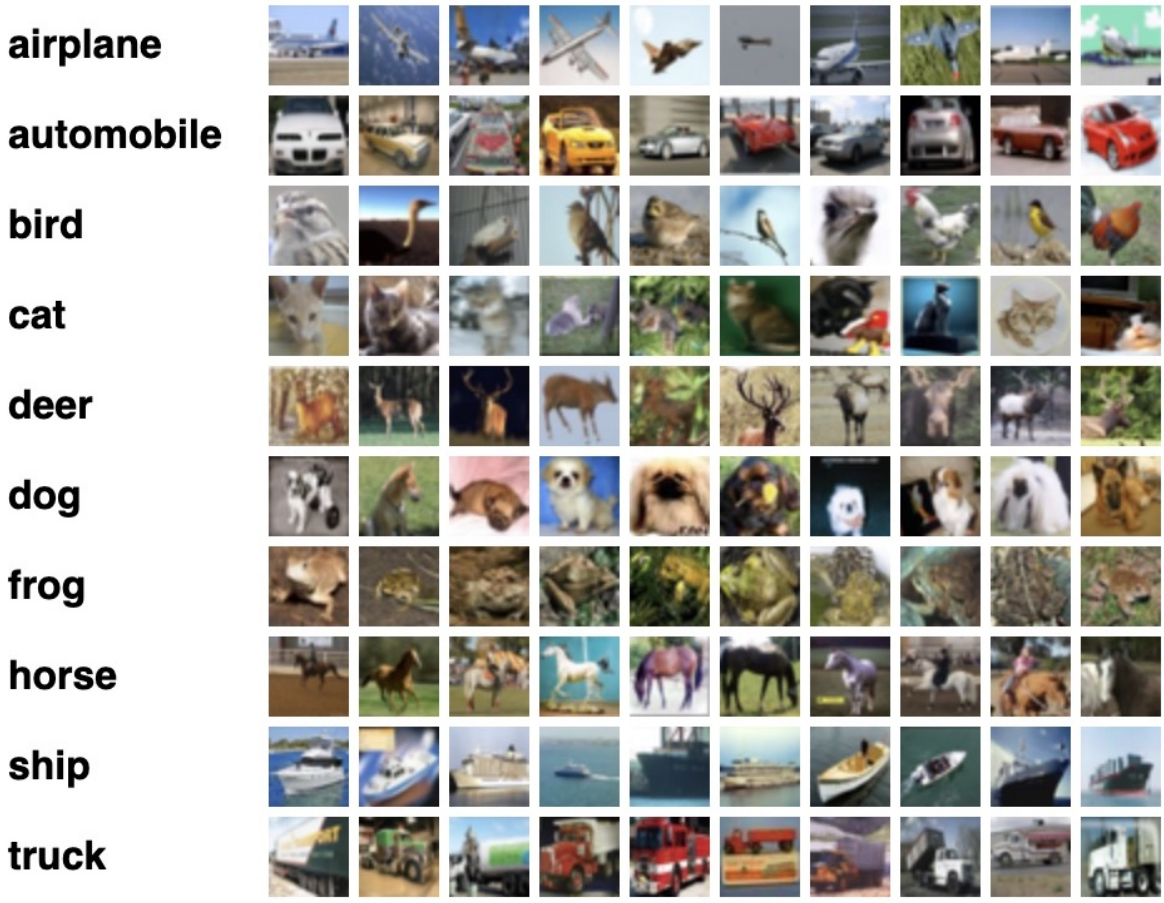


Figura 4.2: Amostra de imagens do conjunto de dados CIFAR-10. Disponível em: `https://www.cs.toronto.edu/~kriz/cifar.html`.

O modelo VGG16 foi treinado nos conjuntos de dados CIFAR-10 e CIFAR-100. O conjunto de dados CIFAR-10 (KRIZHEVSKY, 2009) contém 60.000 imagens coloridas de di-

mensões 32 $\times$ 32 pixels, igualmente divididas em 10 classes, com 6.000 imagens por classe. Destas 60.000 imagens, 50.000 constituem o conjunto de treinamento e 10.000 constituem o conjunto de teste. A Figura 4.2 exibe algumas das imagens que compõem o conjunto de dados CIFAR-10. Já o conjunto de dados CIFAR-100 (KRIZHEVSKY, 2009) tem a mesma quantidade de imagens coloridas, com as mesmas dimensões do CIFAR-10, sendo a única diferença que as imagens estão igualmente distribuídas em 100 classes diferentes ao invés de 10 classes, com cada classe associada a uma das 20 superclasses, conforme a Tabela 4.1. A maior quantidade de classes no conjunto de dados CIFAR-100 implica menor quantidade de imagens por classe - são 6 mil imagens por classe no conjunto de dados CIFAR-10 e 600 imagens por classe no conjunto de dados CIFAR-100 - resultando em um problema de classificação mais desafiador devido à menor quantidade de instâncias por classe.

A utilização desses conjuntos de dados se deu pois o treinamento em conjuntos maiores, como é o caso do ImageNet (RUSSAKOVSKY et al., 2015), implicaria treinamentos mais demorados que são potencializados pelo alto tempo de treinamento inerente à capacidade do modelo VGG16. Logo a utilização de conjuntos de dados menores, como é o caso dos CIFAR-10 e CIFAR-100, viabiliza a realização de um volume maior de experimentos. Ademais, tanto o modelo VGG16 quanto os conjuntos de dados CIFAR-10 e CIFAR-100 são utilizados em pesquisas anteriores para avaliar métodos de poda de redes neurais (LI et al., 2017; LUO; WU; LIN, 2017; HE; ZHANG; SUN, 2017; LIU, Zhuang et al., 2019).

Para ambos os conjuntos de dados, treinou-se o modelo VGG16 por 160 épocas, de acordo com o protocolo de treinamento usado em Zhuang Liu et al. (2019) para os conjuntos de dados CIFAR (KRIZHEVSKY, 2009). Além disso, aplicou-se a abordagem de *data augmentation* padrão proposta por Kaiming He et al. (2016). O método de otimização SGD com *momentum* é usado com uma taxa de aprendizado inicial de 0,1, que decai (dividindo por 10) nas épocas 80 e 120 como em Zhuang Liu et al. (2019).

Os experimentos foram conduzidos em uma máquina rodando Ubuntu 16.04.6 LTS Linux Kernel 4.15.0-107-generic equipada com processador Intel ©Core™i7-8700K CPU @ 3.70GHz, memória RAM 2$\times$16GiB @ 2666 MHz memory e GPU GeForce 2080 RTX Ti. As CNNs e os algoritmos de poda foram implementados usando Python v3.6.9 (VAN ROSSUM; DRAKE, 2009), PyTorch (PASZKE et al., 2019) v1.3.1 e os conjuntos de dados do pacote torchvision v0.4.2. A métrica DeepLIFT foi calculada usando a biblioteca Cap-

Superclasse	**Classes**
aquatic mammals	beaver, dolphin, otter, seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchids, poppies, roses, sunflowers, tulips
food containers	bottles, bowls, cans, cups, plates
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
household electrical devices	clock, computer keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium-sized mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple, oak, palm, pine, willow
vehicles 1	bicycle, bus, motorcycle, pickup truck, train
vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

Tabela 4.1: Divisão do conjunto de dados CIFAR-100 em superclasses e classes. Tabela adaptada de: `https://www.cs.toronto.edu/~kriz/cifar.html`.

tum (KOKHLIKYAN et al., 2020). Além disso, as bibliotecas CUDA v10.1.243 e cuDNN v7.6.3 foram utilizadas para execução das CNNs em GPU.

4.2 Metodologia

A metodologia experimental adotada nesta pesquisa contempla as seguintes etapas:

1. Poda baseada na magnitude dos pesos e o surgimento de bilhetes vencedores usando os métodos de Rebobinamento dos Pesos e Rebobinamento da Taxa de Aprendizagem;

2. Poda baseada no método de interpretabilidade DeepLIFT usando poda global e Rebobinamento da Taxa de Aprendizagem.

Para os experimentos de poda baseada na magnitude dos pesos, avaliou-se o surgimento de "bilhetes vencedores" através da poda estruturada do modelo de rede neural convolucional VGG16 de forma iterativa em dois cenários: poda local e poda global. Para ambos os cenários, a norma-L1 dos pesos dos filtros convolucionais foi utilizada como heurística de poda, removendo-se a cada iteração de poda os filtros com os menores valores de norma, de acordo com uma taxa de poda escolhida, conforme especificado no Algoritmo 1. Para cada procedimento de poda, 10 iterações eram realizadas, aplicando-se uma taxa de poda de 20%, como em pesquisas anteriores (FRANKLE; CARBIN, 2019; MORCOS et al., 2019; RENDA; FRANKLE; CARBIN, 2020). Nesta primeira etapa da metodologia experimental foram utilizadas as seguintes técnicas de retreinamento:

- Rebobinamento dos Pesos, proposta por Frankle e Carbin (2019) e revisada por Frankle et al. (2019);

- Rebobinamento da Taxa de Aprendizagem, proposta por Renda, Frankle e Carbin (2020).

Para os experimentos de poda com base no método de explicabilidade DeepLIFT (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017) se manteve o protocolo da poda iterativa, com 10 iterações e taxa de poda 20%, mas optou-se por utilizar exclusivamente a combinação de poda global e a técnica de retreinamento de Rebobinamento da Taxa de Aprendizagem, pois essa combinação obteve os melhores resultados na primeira etapa da metodologia experimental. A seleção dos filtros a serem removidos foi feita com base na média da norma-L1 dos valores estimados através do método DeepLIFT na partição de treinamento dos conjuntos de dados. Para estimar os valores de DeepLIFT, uma versão embaçada da imagem original foi

utilizada como entrada de referência, conforme especificado no Algoritmo 2, uma vez que esta é a sugestão dos autores para os conjuntos de dados CIFAR-10 e CIFAR-100, conforme descrito no apêndice L do artigo que apresenta o método.

As principais contribuições desta dissertação estão diretamente relacionadas às abordagens de poda utilizadas, uma vez que a pesquisa apresenta um foco exclusivo em poda estruturada para o surgimento de "bilhetes vencedores", abordagem sub-representada em pesquisas anteriores relacionadas à Hipótese do Bilhete de Loteria (FRANKLE; CARBIN, 2019; FRANKLE et al., 2019; MORCOS et al., 2019; ZHOU, H. et al., 2019; RENDA; FRANKLE; CARBIN, 2020). Além disso, a utilização do método DeepLIFT, como critério para seleção dos elementos a serem removidos durante o processo de poda, em combinação com o método de Rebobinamento da Taxa de Aprendizagem representa uma abordagem inédita de acordo com a revisão bibliográfica realizada nesta pesquisa e apresentada no Capítulo 3.

Algoritmo 1: Poda Baseada na Magnitude dos Pesos

PodaMagnitude (m, g, t)

entradas: Modelo treinado m; Localidade de poda l; Taxa de poda t

saída **:** Modelo podado m'

para cada *camada convolucional* $c_i \in m$ **faça**

  para cada *filtro* $f_i \in c_i$ **faça**

    calcular a norma L1 dos pesos de f_i;

  fim

fim

se $l = global$ **então**

  ordenar normas independente da camada;

  podar os filtros com menores normas até alcançar t;

senão se $l = local$ **então**

  ordenar normas por camada;

  podar os filtros com menores normas até alcançar t por camada;

fim

retorna m';

Algoritmo 2: Poda Baseada no Método DeepLIFT

PodaDeepLIFT (m, D, t)

entradas: Modelo treinado m; Conjunto de dados D; Taxa de poda t

saída **:** Modelo podado m'

para cada *camada convolucional* $c_i \in m$ **faça**

para cada *entrada* $e_i \in D$ **faça**

referência $r_i \leftarrow$ aplicar blur em e_i;

calcular os valores de DeepLIFT(e_i, r_i);

fim

para cada *filtro* $f_i \in c_i$ **faça**

calcular a norma L1 dos valores de DeepLIFT de f_i;

fim

fim

ordenar normas independente da camada;

podar os filtros com menores normas até alcançar t;

retorna m';

Capítulo 5

Resultados e Discussão

Neste capítulo, são apresentados e discutidos os resultados dos experimentos desenvolvidos para avaliar a solução proposta, englobando os aspectos descritos em detalhes no Capítulo 4. Na Seção 5.1, são apresentados os resultados da etapa 1 da metodologia experimental, que diz respeito à poda baseada na magnitude dos pesos e o surgimento de bilhetes vencedores usando os métodos de Rebobinamento dos Pesos e Rebobinamento da Taxa de Aprendizagem. Na Seção 5.2, são apresentados os resultados da etapa 2 da metodologia experimental, que diz respeito à poda baseada no método de explicabilidade DeepLIFT. Tanto na Seção 5.1 quanto na Seção 5.2, todos os gráficos que apresentam as curvas de acurácia ao longo das iterações de poda mostram a mediana, o valor máximo e o valor mínimo de acurácia nos conjuntos de teste de cinco rodadas de treinamento e de poda. Além disso, em cada uma das curvas, os bilhetes vencedores com maior acurácia são marcados com losangos e a acurácia da rede não podada também é mostrada como uma linha tracejada.

5.1 Rebobinamento dos Pesos e Rebobinamento da Taxa de Aprendizagem

Nesta seção, são apresentados experimentos que avaliam a possibilidade de surgimento de bilhetes vencedores usando os métodos de retreinamento com Rebobinamento dos Pesos e com Rebobinamento da Taxa de Aprendizagem em um contexto de poda estruturada. Esses experimentos têm a finalidade de verificar se os benefícios da aplicação dos métodos de re-

bobinamento (FRANKLE; CARBIN, 2019; FRANKLE et al., 2019; RENDA; FRANKLE; CARBIN, 2020) também podem ser observados no contexto de poda estruturada de redes neurais convolucionais. Para avaliar a eficácia desses métodos de retreinamento, as curvas de acurácia obtidas por cada abordagem de retreinamento são comparadas com as curvas de acurácia obtidas através do retreinamento com pesos aleatórios das mesmas redes neurais podadas, de modo que haverão redes com a mesma arquitetura resultante do processo de poda, variando-se apenas a abordagem de retreinamento. Além disso, são consideradas também as abordagens de poda global e local, descritas na Subseção 2.2.2.

5.1.1 Rebobinamento dos Pesos

Para avaliar o surgimento de bilhetes vencedores por rebobinamento dos pesos em redes neurais convolucionais podadas através de poda estruturada, utiliza-se a técnica de retreinamento como apresentada em Frankle e Carbin (2019) e Frankle et al. (2019). As épocas de de treinamento 0 a 4 são consideradas como épocas de rebobinamento dos pesos, ou seja, os valores dos pesos em cada uma destas épocas do treinamento inicial da rede original não podada são atribuídos aos respectivos pesos remanescentes das redes neurais podadas durante o processo de retreinamento.

CIFAR-10

Como pode ser observado na Figura 5.1, a abordagem de rebobinamento dos pesos com os pesos aleatórios iniciais (época 0) ou os pesos das primeiras épocas de treinamento (épocas de 1 a 4) da rede não podada não têm um impacto significativo na acurácia final do modelo podado de forma estruturada, tanto para a poda local quanto para a poda global. Em média, as redes podadas obtidas por rebobinamento dos pesos apresentaram menor acurácia do que a rede não podada, mesmo nas primeiras iterações do processo de poda iterativa. Embora poucos, bilhetes vencedores surgiram quando aplicou-se a abordagem de poda global e os pesos da rede podada foram rebobinados para os pesos da época 2 e da época 3 do treinamento da rede não podada. Entretanto, estes bilhetes vencedores surgiram na primeira iteração de poda, quando apenas 20% dos filtros convolucionais foram removidos, de modo que não pode ser considerado um nível de poda agressivo.

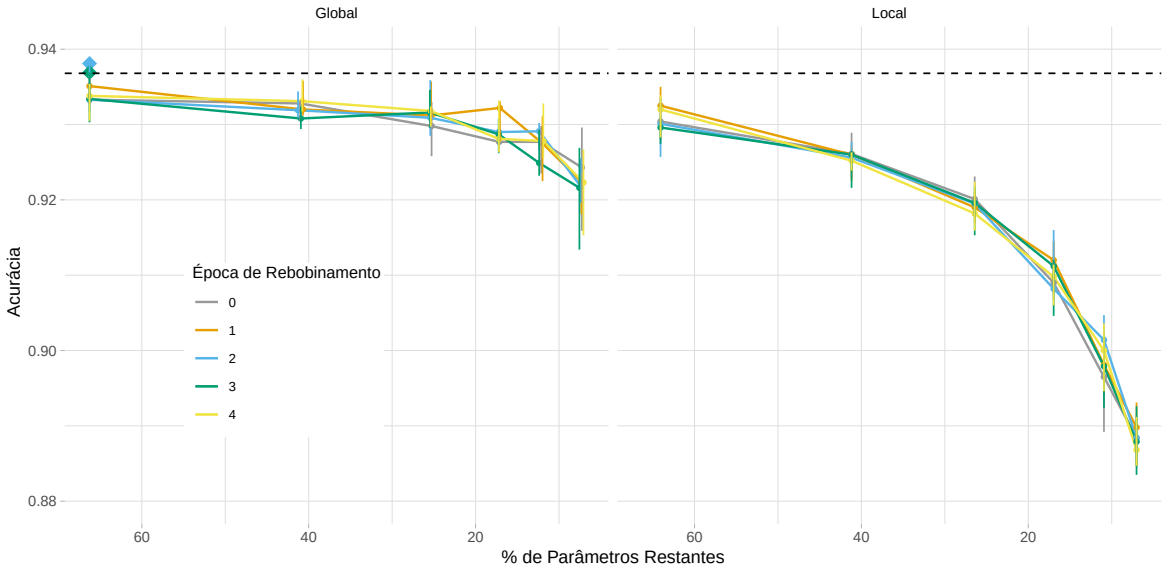


Figura 5.1: Acurácia da CNN VGG16 treinada no conjunto de dados CIFAR-10, podada iterativamente usando rebobinamento de pesos para diferentes épocas.

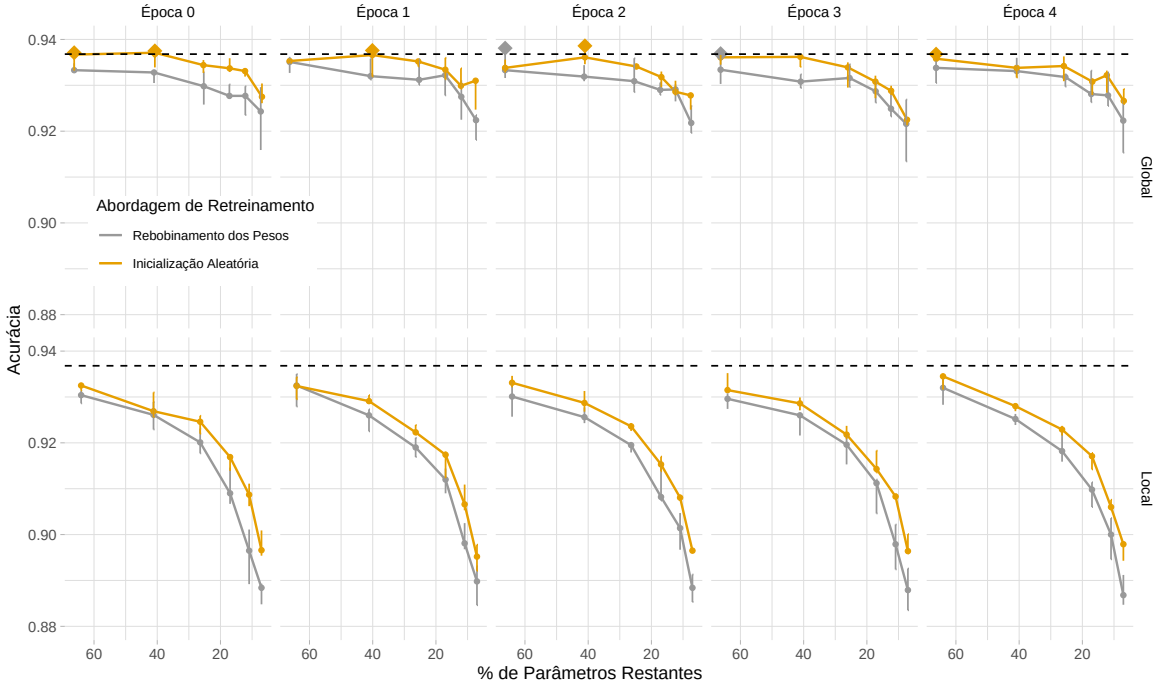


Figura 5.2: Comparação das acurácias obtidas com rebobinamento dos pesos e com inicialização aleatória dos pesos da CNN VGG16 treinada no conjunto de dados CIFAR-10, podada iterativamente usando rebobinamento de pesos para diferentes épocas.

Pode-se observar ainda na Figura 5.1 que, embora a poda global gere redes podadas que geralmente apresentam maior acurácia que as redes podadas geradas pela poda local, a poda global é menos estável que a poda local e apresenta uma maior variação na acurácia obtida em níveis de poda mais agressivos (ou seja, quando a porcentagem de parâmetros restantes na rede podada é menor). Isto se deve ao fato de que a poda global em combinação com a poda estruturada pode causar a remoção de muitos filtros convolucionais em camadas específicas, o que dificulta com que a rede podada aprenda a regra de classificação ao extrair um número insuficiente de características da imagem de entrada.

Na Figura 5.2, são comparadas as curvas de acurácia geradas pelas redes podadas obtidas com o rebobinamento dos pesos com suas contrapartes retreinadas usando pesos inicializados aleatoriamente. Podemos observar que, para quase todas as redes de sub-rede geradas, aquelas treinadas com pesos aleatórios apresentam acurácia igual ou superior a das redes podadas obtidas por rebobinamento dos pesos. Isto pode ser observado tanto quando os pesos são rebobinados para os pesos iniciais aleatórios da rede não podada (época 0), como quando os pesos são rebobinados para os pesos das primeiras épocas do treinamento da rede não podada (épocas de 1 a 4), tanto na poda global quanto na poda local. Além disso, ainda é possível notar uma maior ocorrência de bilhetes vencedores (pontos em forma de losango nos subgráficos) nas redes podadas retreinadas com pesos aleatórios.

CIFAR-100

Na Figura 5.3, que apresenta os resultados da aplicação do rebobinamento dos pesos no conjunto de dados CIFAR-100, tem-se um resultado similar ao que acontece no CIFAR-10, onde a abordagem de rebobinamento dos pesos não tem um impacto significativo na acurácia final do modelo podado de forma estruturada, tanto para a poda local quanto para a poda global. Uma única diferença é perceptível, pois bilhetes vencedores surgiram nas duas primeiras iterações de poda quando aplicou-se a abordagem de poda global no CIFAR-100, enquanto no CIFAR-10 os bilhetes vencedores só surgiram na primeira iteração de poda.

Na Figura 5.4, que é análoga à Figura 5.2 com a diferença de que é usado o conjunto de dados CIFAR-100, pode-se observar que diferente do que acontece no CIFAR-10, há uma sobreposição das curvas de acurácia das duas abordagens de retreinamento. Este resultado indica que não há benefício ao se aplicar a abordagem de rebobinamento dos pesos. Além

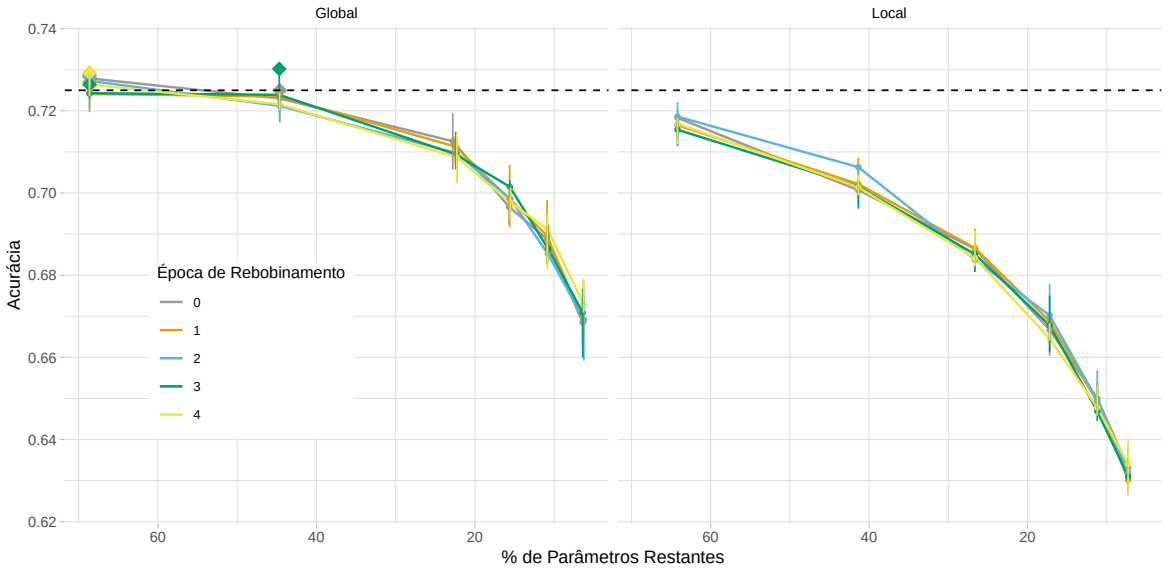


Figura 5.3: Acurácia da CNN VGG16 treinada no conjunto de dados CIFAR-100, podada iterativamente usando rebobinamento de pesos para diferentes épocas.

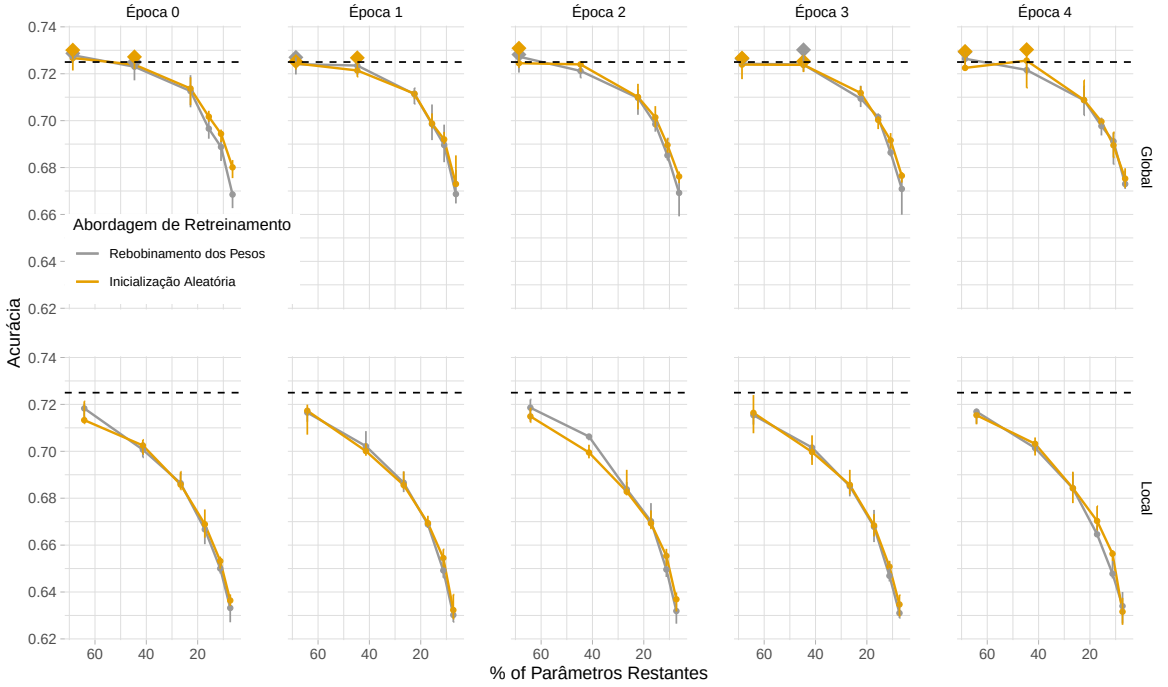


Figura 5.4: Comparação das acurácias obtidas com rebobinamento dos pesos e com inicialização aleatória dos pesos da CNN VGG16 treinada no conjunto de dados CIFAR-100, podada iterativamente usando rebobinamento de pesos para diferentes épocas.

disso, ainda é possível notar a ocorrência simultânea de bilhetes vencedores (pontos em forma de losango nos subgráficos) nas duas abordagens de retreinamento.

Considerações

Os resultados apresentados nesta subseção sugerem que, apesar dos resultados sem precedentes alcançados em redes podadas de forma não-estruturada (FRANKLE; CARBIN, 2019; FRANKLE et al., 2019), o Rebobinamento dos Pesos não é eficiente para encontrar bilhetes vencedores em redes podadas de forma estruturada e apresenta um desempenho inferior àquele obtido pelo retreinamento das redes podadas com pesos inicializados aleatoriamente.

5.1.2 Rebobinamento da Taxa de Aprendizagem

Para avaliar o surgimento de bilhetes vencedores por rebobinamento da taxa de aprendizagem em redes neurais convolucionais podadas através de poda estruturada, utiliza-se a técnica de retreinamento como apresentada em Renda, Frankle e Carbin (2020).

CIFAR-10

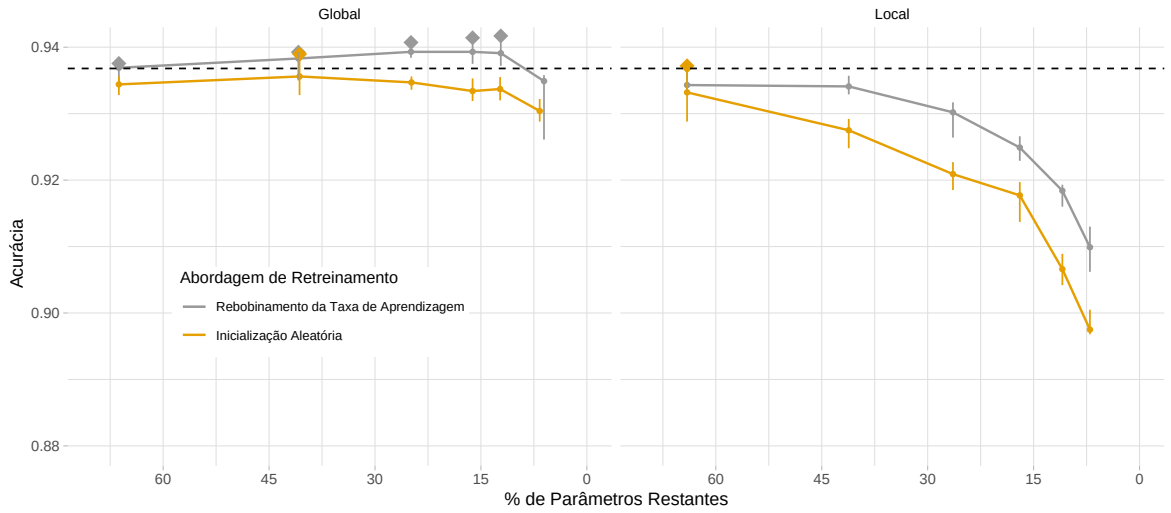


Figura 5.5: Comparação das acurácias obtidas com rebobinamento da taxa de aprendizagem e com inicialização aleatória dos pesos da CNN VGG16 treinada no conjunto de dados CIFAR-10, enquanto é podada iterativamente.

Como pode ser observado na Figura 5.5, diferentemente da abordagem de rebobinamento dos pesos, ao se aplicar o rebobinamento da taxa de aprendizagem, pode-se notar um surgimento maior de bilhetes vencedores, inclusive em iterações posteriores de poda, ou seja, para níveis de poda mais agressivos.

Enquanto que com o rebobinamento dos pesos só foi possível identificar redes podadas que se aproximavam da acurácia da rede original não podada, os bilhetes vencedores gerados pelo rebobinamento da taxa de aprendizagem apresentam maior acurácia que a rede não podada. Esse fenômeno ocorre mesmo em redes podadas que possuem menos de 85% dos parâmetros restantes, quando comparado com a rede original não podada. No entanto, os bilhetes vencedores surgiram unicamente por intermédio da abordagem de poda global.

Pode-se observar ainda na Figura 5.5 que, como ocorreu com o rebobinamento dos pesos, as redes podadas com a poda global tendem a ser menos estáveis, apresentando maior variação na acurácia final entre as diferentes rodadas de poda e treinamento em níveis de poda mais agressivos.

CIFAR-100

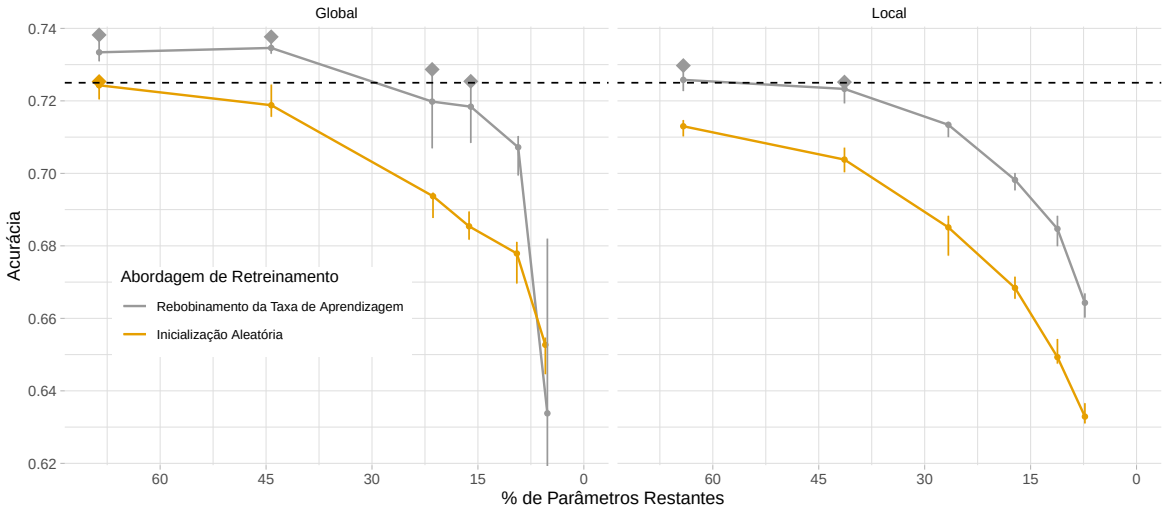


Figura 5.6: Comparação das acurácias obtidas com rebobinamento da taxa de aprendizagem e com inicialização aleatória dos pesos da CNN VGG16 treinada no conjunto de dados CIFAR-100, enquanto é podada iterativamente.

Os resultados observados na Figura 5.6, onde aplicou-se a abordagem de rebobinamento

da taxa de aprendizagem no conjunto de dados CIFAR-100, são similares aos observados na Figura 5.5. No entanto, vale ressaltar algumas particularidades:

1. A diferença entre as curvas de acurácia é maior que a observada no CIFAR-10, com uma vantagem ainda maior para a abordagem de rebobinamento da taxa de aprendizagem;

2. A partir da quinta iteração de poda, quando aplicou-se a abordagem de poda global, foi observada uma degradação maior da acurácia. Este fenômeno pode estar relacionado ao fato do conjunto de dados CIFAR-100 contemplar uma tarefa de classificação mais difícil que aquela do conjunto de dados CIFAR-10, sendo mais sensível à remoção de filtros convolucionais, especialmente em níveis de poda mais agressivos;

3. Por fim, temos o surgimento de bilhetes vencedores nas duas primeiras iterações de poda, quando aplicou-se a abordagem de poda local.

Considerações

Apesar dos experimentos utilizarem uma arquitetura (VGG-16 (SIMONYAN; ZISSERMAN, 2015)) diferente daquelas utilizadas por Renda, Frankle e Carbin (2020) (ResNet-56 e ResNet-34 (HE, K. et al., 2016)), todas essas arquiteturas são redes neurais convolucionais. Levando isso em consideração, nesta dissertação foram encontrados bilhetes vencedores em níveis de compressão mais agressivos que os bilhetes vencedores encontrados pelos autores em redes podadas de forma estruturada.

Comparando com os resultados da pesquisa de Renda, Frankle e Carbin (2020), eles aplicam poda estrutura com rebobinamento da taxa de aprendizagem para uma pequena faixa de níveis de compressão, de aproximadamente 86.96% a 58.82% de parâmetros restantes na rede ResNet-56 e 92.60% a 79.36% de parâmetros restantes na rede ResNet-34, ambas treinadas no conjunto de dados CIFAR-10 (KRIZHEVSKY, 2009). Os bilhetes vencedores identificados pelos autores surgiram em níveis baixos de compressão ($\approx 86.96\%$ e $\approx 77.52\%$ dos parâmetros restantes na rede ResNet-56 e $\approx 92.60\%$ e $\approx 86.96\%$ na rede ResNet-34)[1], em um regime de poda pouco agressivo. Por outro lado, nesta pesquisa considerou-se uma

[1]Os percentuais de parâmetros restante foram calculados com base nos valores de taxa de compressão apresentados na pesquisa de Renda, Frankle e Carbin (2020).

faixa maior de níveis de compressão, de aproximadamente 66.27% a 1.02% de parâmetros restantes na rede VGG-16 treinada no conjunto de dados CIFAR-10 e os bilhetes vencedores foram identificados em níveis altos de compressão ($\approx 66.27\%$, $\approx 40.85\%$, $\approx 24.90\%$, $\approx 16.18\%$ e $\approx 12.19\%$ de parâmetros restantes). Em conclusão, a avaliação experimental da abordagem de rebobinamento da taxa de aprendizagem apresentada nesta dissertação cobriu uma faixa mais ampla de níveis de compressão e foi capaz de identificar bilhetes vencedores em níveis de compressão mais agressivos (ou seja, mais adequados para computação em hardware com recursos limitados).

Vale notar ainda que as redes podadas obtidas através do rebobinamento da taxa de aprendizagem, em ambos os conjuntos de dados CIFAR-10 e CIFAR-100, apresentam maior acurácia quando comparadas com suas contrapartes retreinadas com pesos inicializados aleatoriamente. Este resultado indica que a técnica de rebobinamento da taxa de aprendizagem tem um impacto relevante na acurácia final das redes podadas de forma estruturada.

5.1.3 Considerações Finais

Os resultados apresentados nesta seção estão relacionados à Questão de Pesquisa 1, definida na Seção 1.2. Tomando-se como evidência o surgimento de bilhetes vencedores, em especial quando foram empregados o método de Rebobinamento da Taxa de Aprendizagem de Renda, Frankle e Carbin (2020) e a abordagem de poda global, pode-se afirmar que a aplicação das técnicas de rebobinamentos em redes neurais convolucionais podadas de forma estruturada possibilita o surgimento de bilhetes vencedores.

5.2 Magnitude dos Pesos e o Método DeepLIFT

Nesta seção é apresentada uma abordagem alternativa à utilização da magnitude dos pesos na qual substitui-se a magnitude dos pesos pelo método de explicabilidade DeepLIFT (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017) como critério de seleção, removendo assim os filtros que apresentem as menores normas L1 dos valores de DeepLIFT. Neste conjunto de experimentos adotou-se o Rebobinamento da Taxa de Aprendizagem junto à Poda Global, combinação essa que produziu os melhores resultados nos experimentos apresentados na Seção 5.1.

5.2.1 CIFAR-10

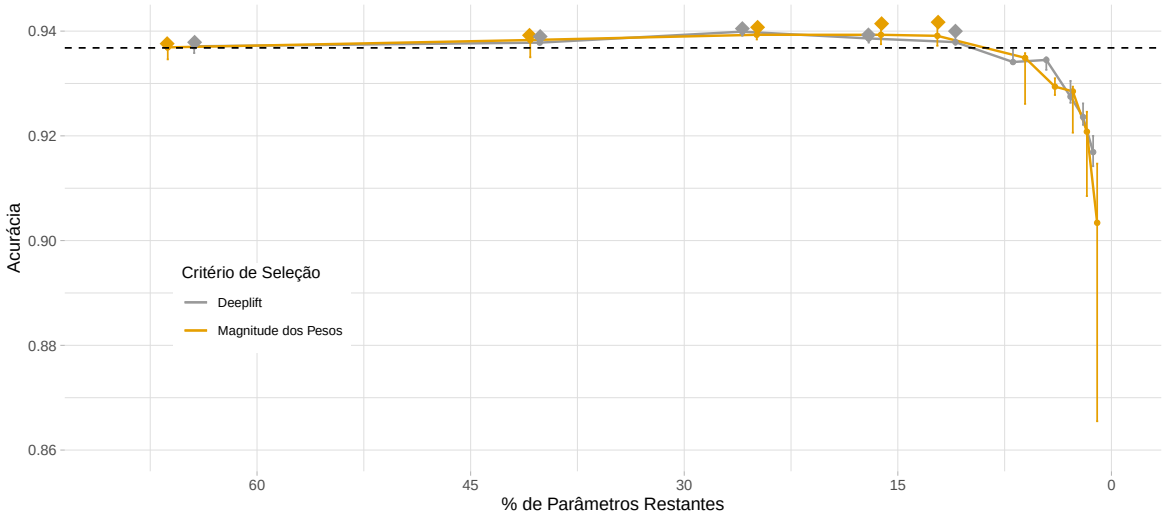


Figura 5.7: Comparação das acurácias obtidas com magnitude dos pesos e DeepLIFT da CNN VGG16 treinada no conjunto de dados CIFAR-10 e podada iterativamente.

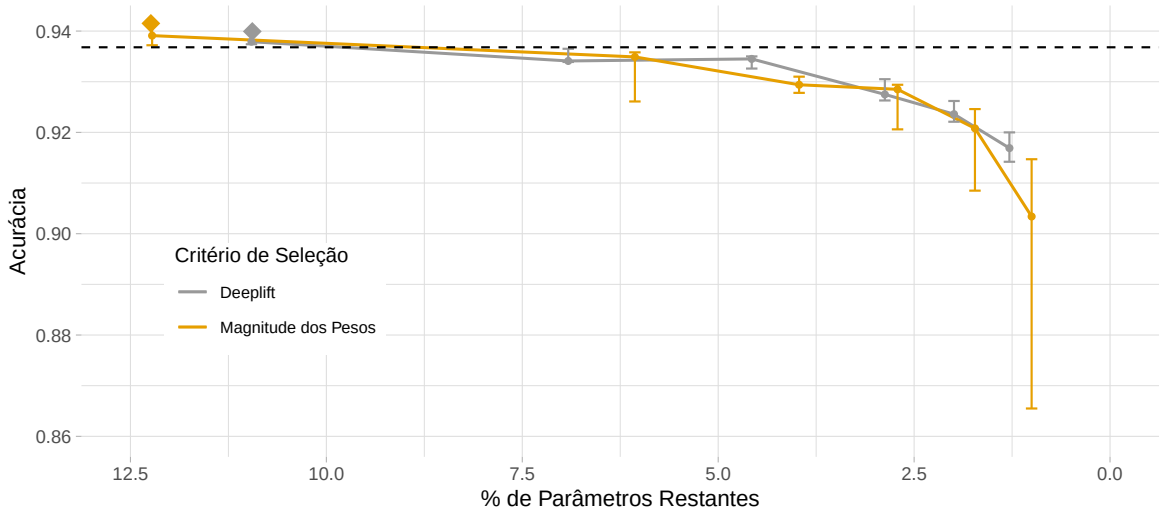


Figura 5.8: Comparação das acurácias obtidas nas iterações finais com magnitude dos pesos e DeepLIFT da CNN VGG16 treinada no conjunto de dados CIFAR-10 e podada iterativamente.

Considerando a rede VGG-16 treinada no conjunto de dados CIFAR-10, pode-se observar na Figura 5.7 que ambos os critérios de seleção foram capazes de gerar a mesma quantidade de bilhetes vencedores, entretanto é possível verificar que as redes podadas ge-

radas usando o critério de seleção baseado na métrica DeepLIFT são mais estáveis, isto é, apresentam uma menor variação dos valores de acurácia entre as diferentes rodadas de treinamento e poda. Além disso, em níveis de poda mais agressivos, as redes podadas usando a métrica DeepLIFT apresentam menor degradação dos valores de acurácia, conforme pode ser melhor observado na Figura 5.8 que foca nas iterações finais de poda.

5.2.2 CIFAR-100

Os resultados observados quando utilizada a rede VGG-16 treinada no conjunto de dados CIFAR-100, segue o mesmo padrão observado nos resultados apresentados nas Subseção 5.2.1, em que ambos os critérios de seleção foram capazes de gerar a mesma quantidade de bilhetes vencedores, como pode ser observado na Figura 5.9. As vantagens do critério de seleção baseado na métrica DeepLIFT apontadas anteriormente são ainda mais evidentes no conjunto de dados CIFAR-100, onde nota-se um claro distanciamento das curvas de acurácia. Na Figura 5.10, que foca nas iterações finais de poda, é possível perceber que, mesmo em níveis de poda extremamente agressivos, o método DeepLIFT produz redes podadas mais estáveis e um ganho médio de 8% em acurácia na última iteração de poda quando comparado com o critério baseado nas magnitude dos pesos.

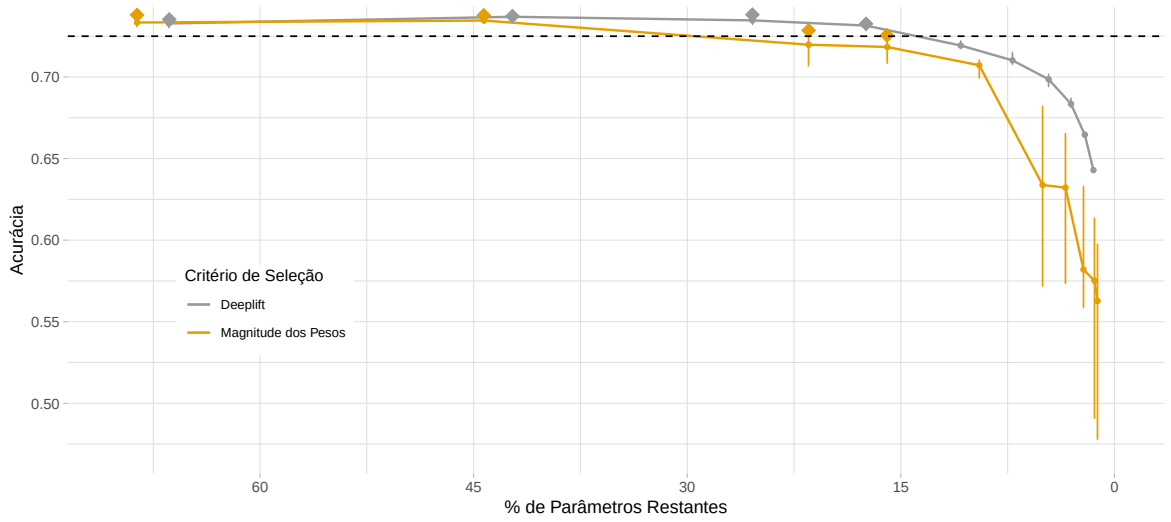


Figura 5.9: Comparação das acurácias obtidas com magnitude dos pesos e DeepLIFT da CNN VGG16 treinada no conjunto de dados CIFAR-100, enquanto é podada iterativamente.

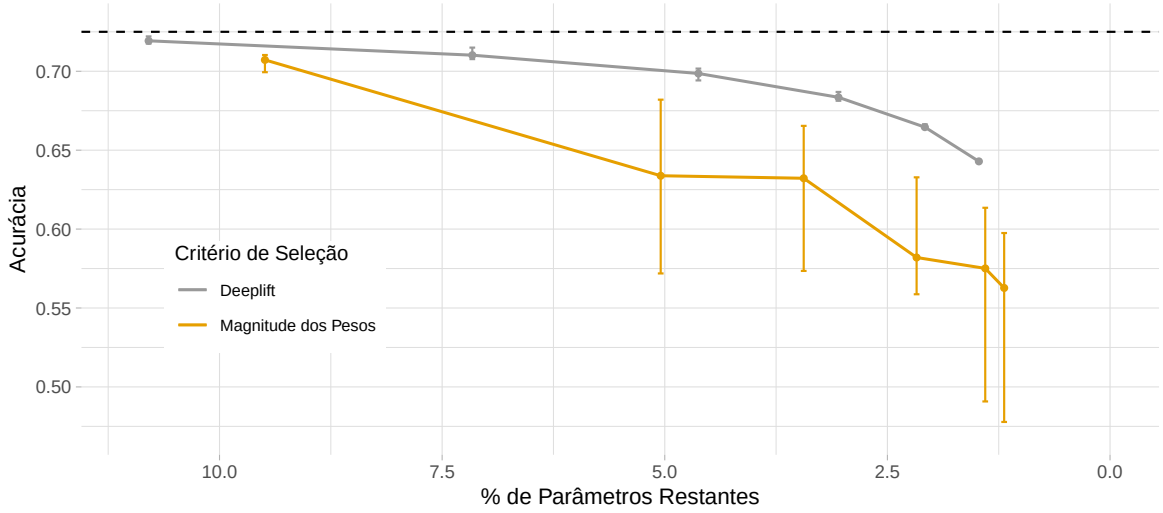


Figura 5.10: Comparação das acurácias obtidas nas iterações finais com magnitude dos pesos e DeepLIFT da CNN VGG16 treinada no conjunto de dados CIFAR-100, enquanto é podada iterativamente.

5.2.3 CNNs Agressivamente Podadas com DeepLIFT

Como já pontuado anteriormente na subseção 2.2.2, a poda global pode inviabilizar a utilização de uma rede neural em casos onde todos - ou quase todos - os pesos, neurônios ou filtros são completamente removidos de uma determinada camada. Este fenômeno está relacionado ao nível de compressão aplicado à rede podada, entretanto pode ser potencializado ou atenuado de acordo com o critério de seleção dos elementos a serem removidos durante o processo de poda.

Nos experimentos realizados e apresentados anteriormente na Seção 5.2, ao aplicar poda global baseada na magnitude dos pesos, aplicou-se 10 iterações de poda à CNN VGG-16, pois esse foi, no geral, o que conseguiu-se podar[2] quando aplicando uma taxa de poda de 20%. Entretanto, utilizando a poda global baseada no método DeepLIFT foi possível aplicar 20 iterações de poda sem que nenhuma camada tivesse todos os filtros convolucionais removidos. Não foi testado um limite de iterações superior a 20, uma vez que a acurácia das redes podadas já havia sido bastante degradada a ponto de não justificar iterações posteriores.

[2]Por se tratar de um método não-determinístico, em algumas execuções do processo de poda o número de iterações foi menor, com alguma camada sendo totalmente removida antes da 10ª iteração e em raras exceções foi maior, com alguma camada sendo totalmente removida após a 11ª iteração.

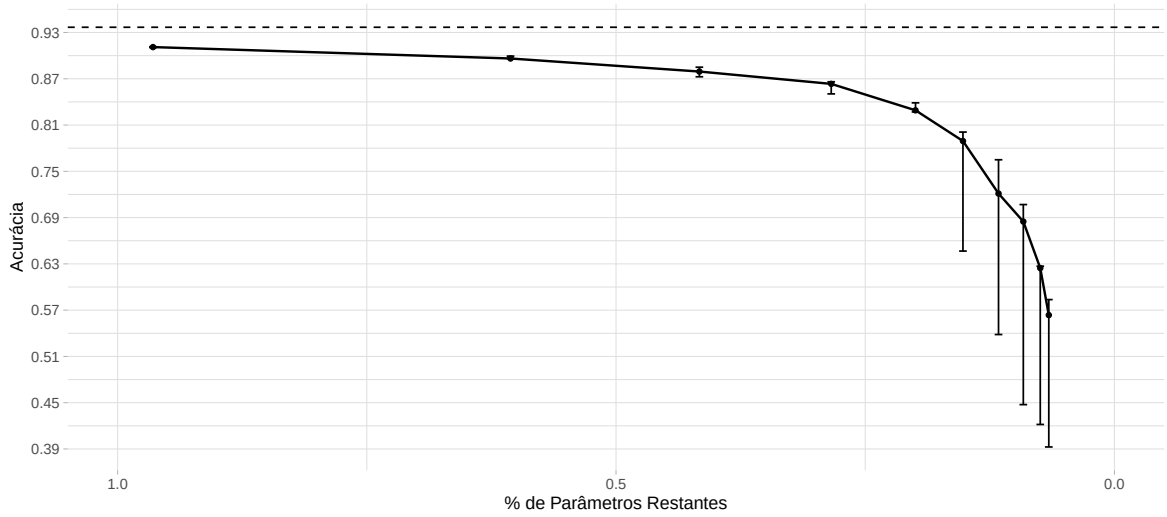


Figura 5.11: Acurácias obtidas em 10 iterações adicionais (iterações 11 a 20) com DeepLIFT da CNN VGG16 treinada no conjunto de dados CIFAR-10, enquanto é podada iterativamente.

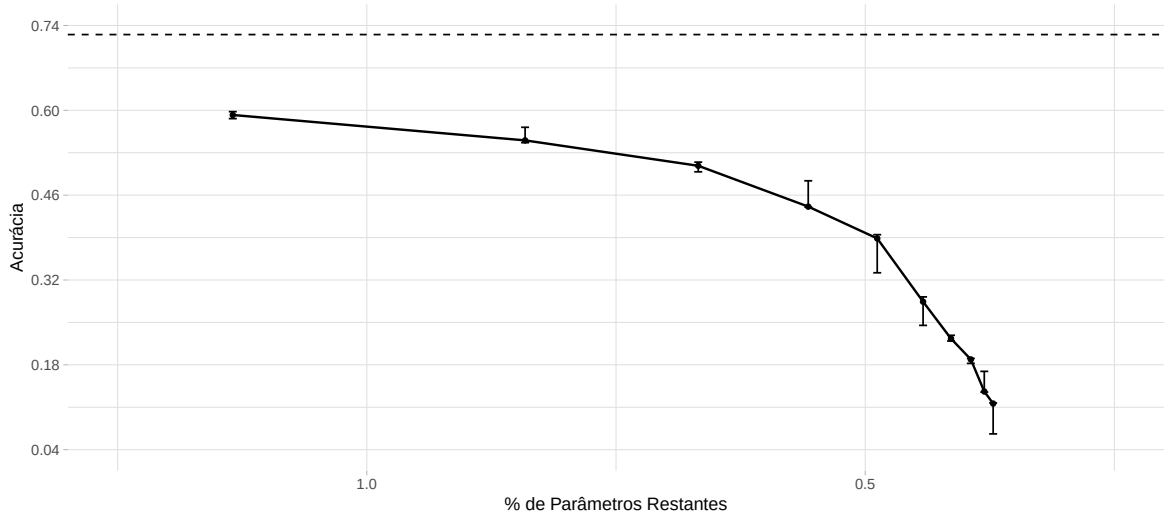


Figura 5.12: Acurácias obtidas em 10 iterações adicionais (iterações 11 a 20) com DeepLIFT da CNN VGG16 treinada no conjunto de dados CIFAR-100, enquanto é podada iterativamente.

Nas Figuras 5.11 e 5.12 pode-se observar as curvas de acurácia das 10 iterações de poda adicionais (iterações 11 a 20) da CNN VGG16 para os conjuntos de dados CIFAR-10 e CIFAR-100. É importante pontuar que, nenhuma das redes produzidas nas iterações adicionais é um bilhete vencedor, pois apresentam valores de acurácia inferiores ao da rede origi-

nal não podada. Por outro lado, considerando os resultados do conjunto de dados CIFAR-10 apresentados na Figura 5.11 foi possível gerar redes podadas com menos de 0,25% dos parâmetros restantes e com uma degradação de menos de 10% da acurácia com relação a rede original não podada. No entanto, também é possível perceber uma maior instabilidade nas interações finais de poda, isto é, maior variação nos valores observados de acurácia.

A capacidade da poda global baseada no método DeepLIFT de alcançar níveis de compressão maiores que a poda global baseada na magnitude dos pesos e ainda manter a estabilidade da rede neural pode estar relacionada com a distribuição por camada dos filtros removidos a cada iteração de poda. Pode-se observar nas Figuras 5.13 e 5.14 que, ao variar o critério de seleção, a forma como os filtros são removidos entre interações consecutivas muda. Na poda global baseada na magnitude dos pesos, o ranqueamento faz com que a poda seja mais concentrada em uma mesma camada a cada iteração, enquanto que na poda global baseada no método DeepLIFT, a remoção dos filtros convolucionais acontece de forma mais distribuída em algumas camadas da CNN. Tomando a Figura 5.13 como exemplo, note que quando comparamos as iterações 5 e 6 da poda baseada na magnitude dos pesos, é possível observar que na iteração 6 os filtros foram podados principalmente da camada 6, camada esta que estava praticamente intocada na iteração 5. O mesmo acontece quando comparamos as iterações 6 e 7, só que agora com os filtros sendo podados da camada 5. Por outro lado, a poda baseada no método DeepLIFT distribui a poda entre as camadas 5, 6 , 7 e 8 ao longo das iterações. Os gráficos referentes a todas as iterações de poda nos conjuntos de dados CIFAR-10 e CIFAR-100 estão no Apêndice A desta dissertação.

5.2.4 Análise do Tempo de Inferência dos Modelos Podados

Nas Figuras 5.15 e 5.16 são apresentados os resultados do *Speedup* médio do tempo de inferência, variando o tamanho do *batch* e a unidade de processamento, na partição de testes dos conjuntos de dados CIFAR-10 e CIFAR-100. O *Speedup* é uma medida que representa o desempenho relativo de dois sistemas/soluções processando o mesmo problema e é calculado de acordo com a Fórmula 5.1 quando usado para medir melhora na latência (tempo de execução) de uma solução, onde $S_{latência}$ é o *speedup* de latência da nova solução em relação a antiga, L_{antiga} é a latência da solução antiga e L_{nova} é a latência da solução nova (HENNESSY; PATTERSON, 2017). No caso específico desta análise, considerou-se L_{antiga} como

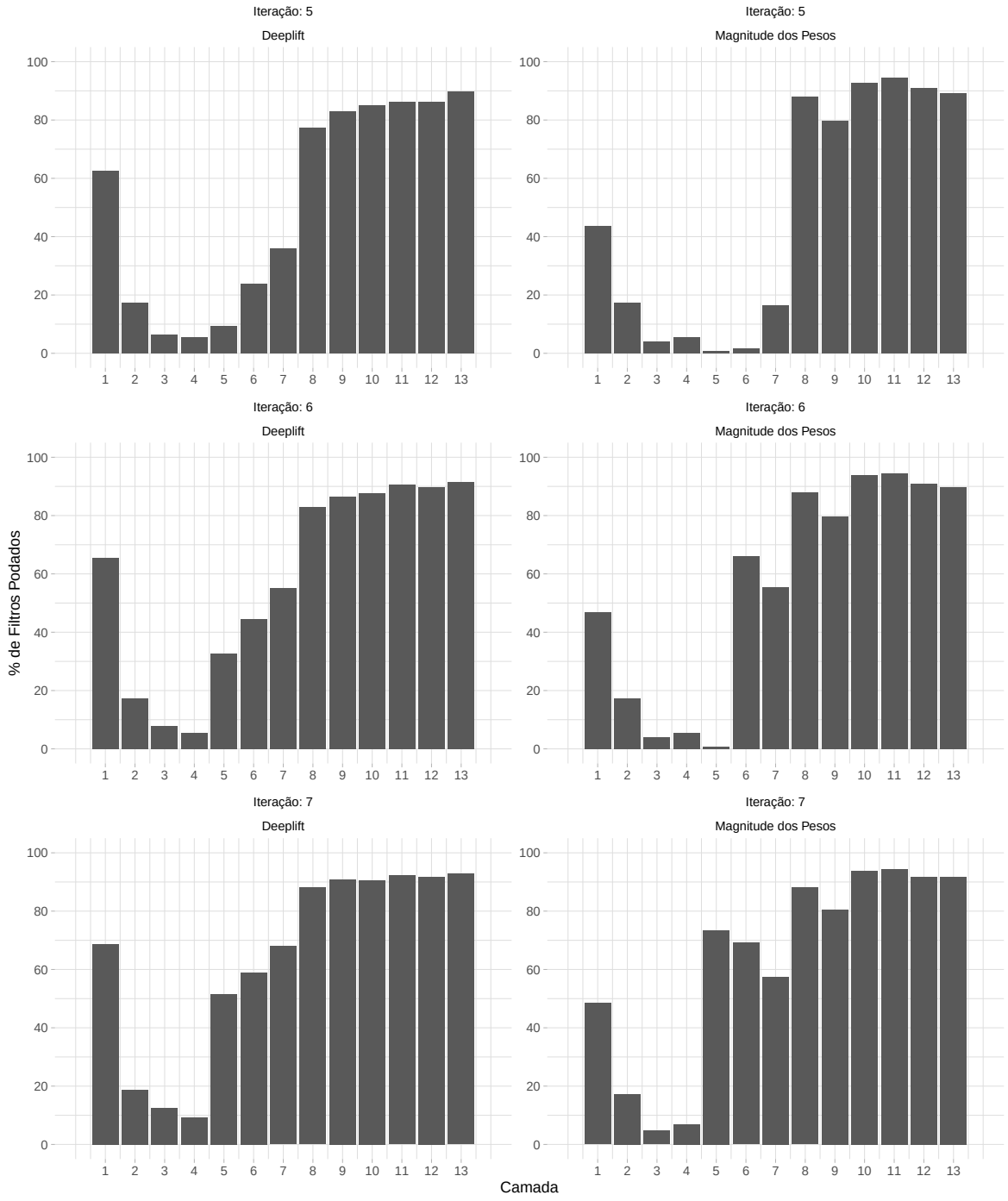


Figura 5.13: Percentual de filtros convolucionais podados com DeepLIFT e com Magnitude dos Pesos nas iterações de poda 5, 6 e 7 da CNN VGG16 no conjunto de dados CIFAR-10.

o tempo de inferência da CNN não-podada e L_{nova} como o tempo de inferência das CNNs podadas.

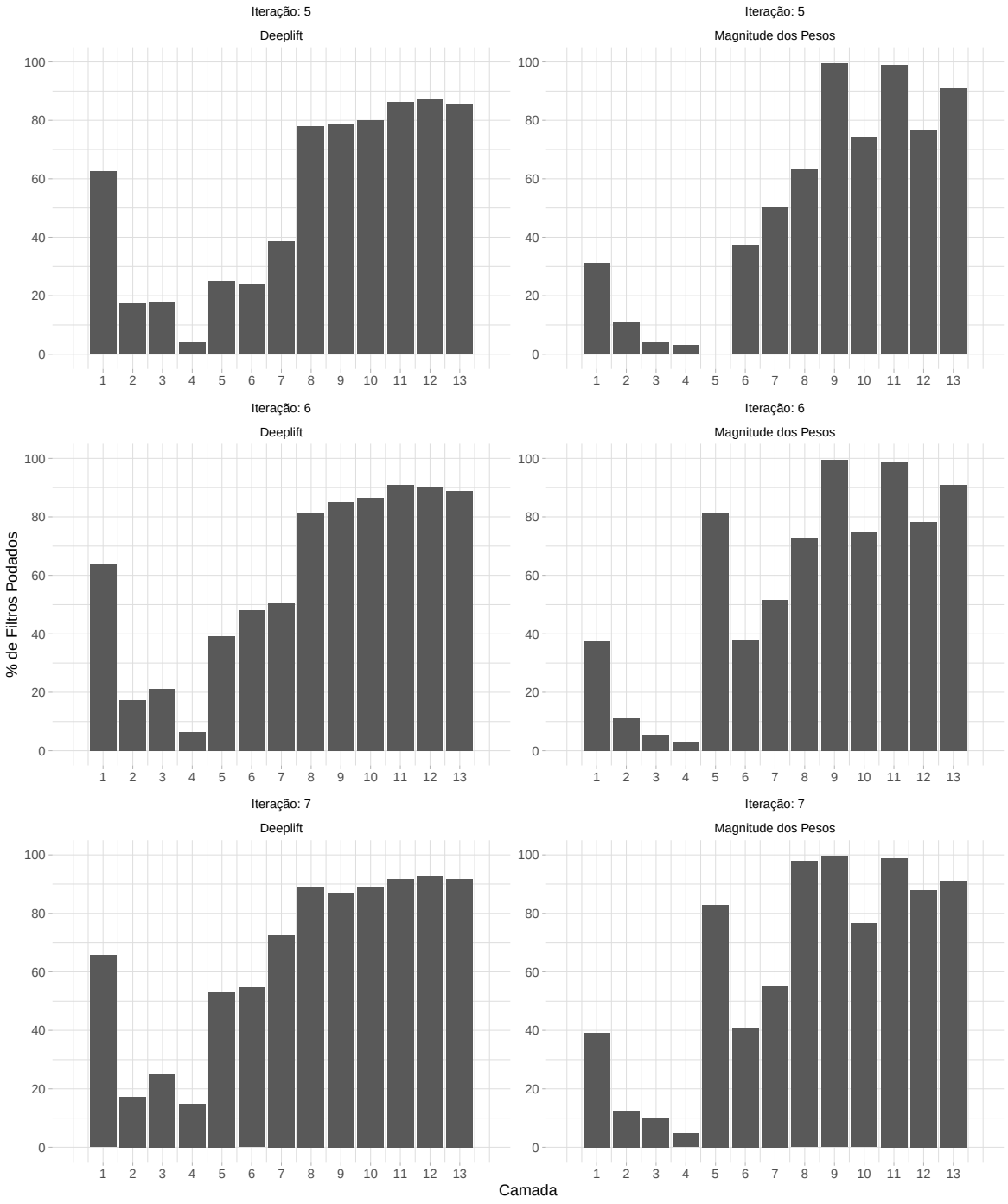


Figura 5.14: Percentual de filtros convolucionais podados com DeepLIFT e com Magnitude dos Pesos nas iterações de poda 5, 6 e 7 da CNN VGG16 no conjunto de dados CIFAR-10.

$$S_{latência} = \frac{L_{antiga}}{L_{nova}} \tag{5.1}$$

O tempo de inferência considerado para o cálculo do *speedup* foi a média dos tempos

de inferência de 10 execuções distintas, sempre com uma execução preliminar - não contabilizada - para mitigar possíveis distorções causadas por operações de leitura e escrita dos dados. Para esta análise, a medição dos tempos de inferência foi realizada em um laptop Dell G5 5590 com 16 GiB de memória RAM SODIMM DDR4 @ 2667 MHz, CPU Hexa-Core Intel©Core™ i7-9750H CPU @ 2.60GHz, GPU NVIDIA GeForce RTX 2060 Mobile com 1920 CUDA *cores*, 240 Tensor *cores*, 6 GiB GDDR6 de memória de vídeo e Linux Mint 20.2 Cinnamon Kernel v5.4.0-90-generic como sistema operacional. O software utilizado foi o mesmo descrito na Seção 4.1.

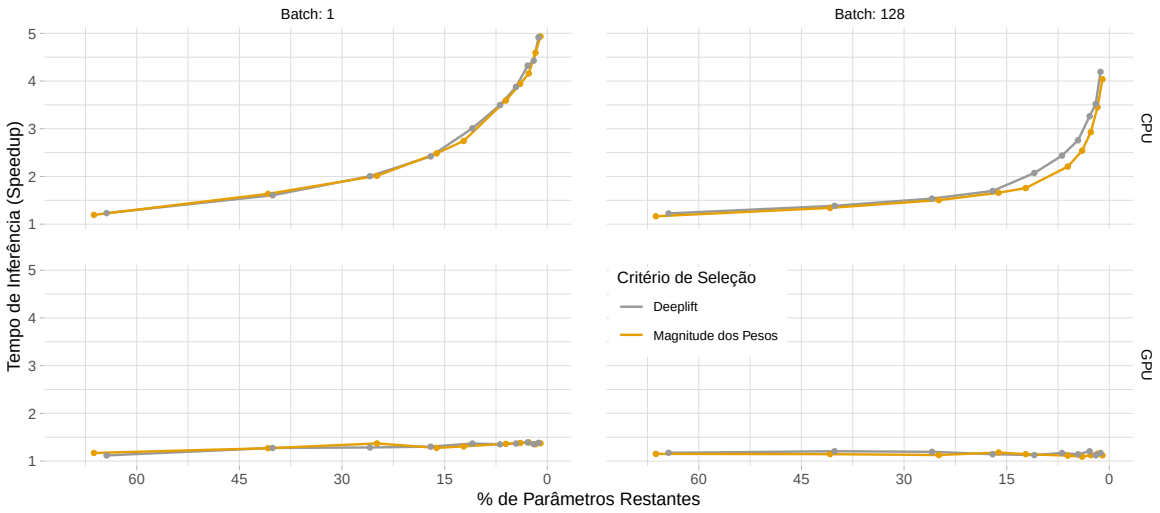


Figura 5.15: *Speedup* do tempo de inferência na partição de teste do conjunto de dados CIFAR-10 das redes podadas com magnitude dos pesos e DeepLIFT obtidas a partir da CNN VGG16.

É possível observar um mesmo padrão que se repete nas Figuras. Por um lado, considerando a execução das CNNs em GPU, o *speedup* é inferior a $1,5\times$ mesmo em redes agressivamente podadas (iterações finais de poda), independente do tamanho do *batch*. Por outro lado, é possível observar um ganho grande em *speedup* quando as redes podadas são executadas em CPU, com *speedups* próximos a $5\times$ para *batches* de tamanho 1 e próximos a $4\times$ para batches de tamanho 128 em redes agressivamente podadas (iterações finais de poda). Vale notar também que, na maioria dos casos, as curvas de *speedup* em relação ao percentual de parâmetros restantes estão sobrepostas indicando que o impacto das duas abordagens de poda no tempo de inferência é similar, com exceção do *batch* de tamanho 128 que

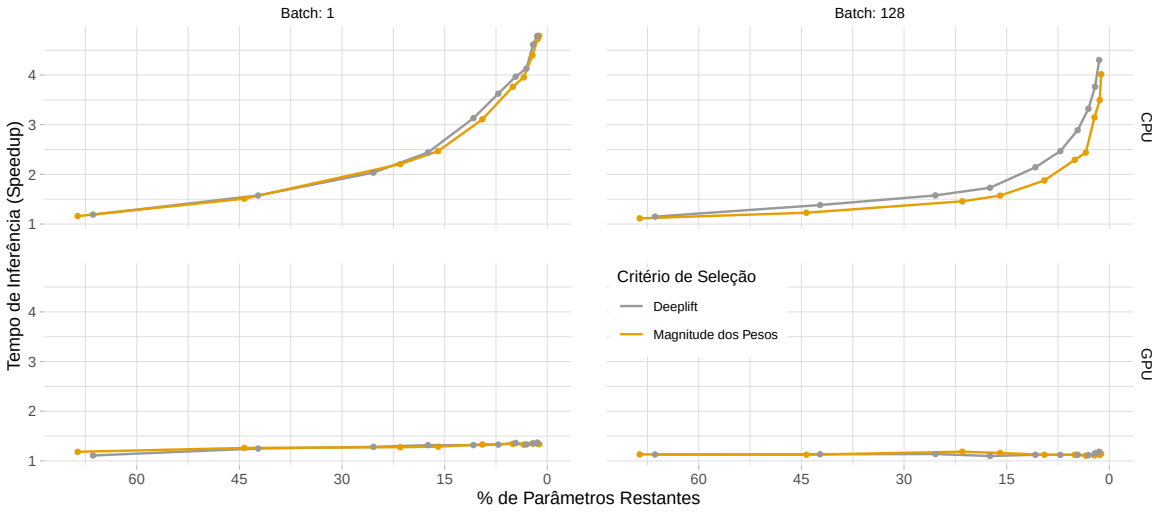


Figura 5.16: *Speedup* do tempo de inferência na partição de teste do conjunto de dados CIFAR-100 das redes podadas com magnitude dos pesos e DeepLIFT obtidas a partir da CNN VGG16.

evidencia uma leve superioridade da abordagem de poda baseada no método DeepLIFT. De forma complementar ao *speedup*, as Figuras 5.17 e 5.18 apresentam os tempos de inferência em segundos.

5.2.5 Considerações Finais

Os resultados apresentados nesta seção estão relacionados à Questão de Pesquisa 2, definida na Seção 1.2. Apesar da utilização do critério de seleção baseado na métrica DeepLIFT (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017) não aumentar a quantidade de bilhetes vencedores identificados, os resultados dos experimentos e análises apresentadas evidenciam que a substituição da magnitude dos pesos pela métrica DeepLIFT melhora a acurácia das redes podadas e torna o processo de poda mais estável, principalmente para níveis de poda mais agressivos. Entretanto, é importante apontar que a aplicação do método DeepLIFT é mais custosa computacionalmente que a Magnitude dos Pesos, uma vez que é necessário calcular os valores de DeepLIFT com base nas entradas usadas para avaliar a importância das características dos filtros, enquanto a magnitude dos pesos é calculada com base nos valores de pesos, sem necessidade de computação adicional além da magnitude e da norma L1.

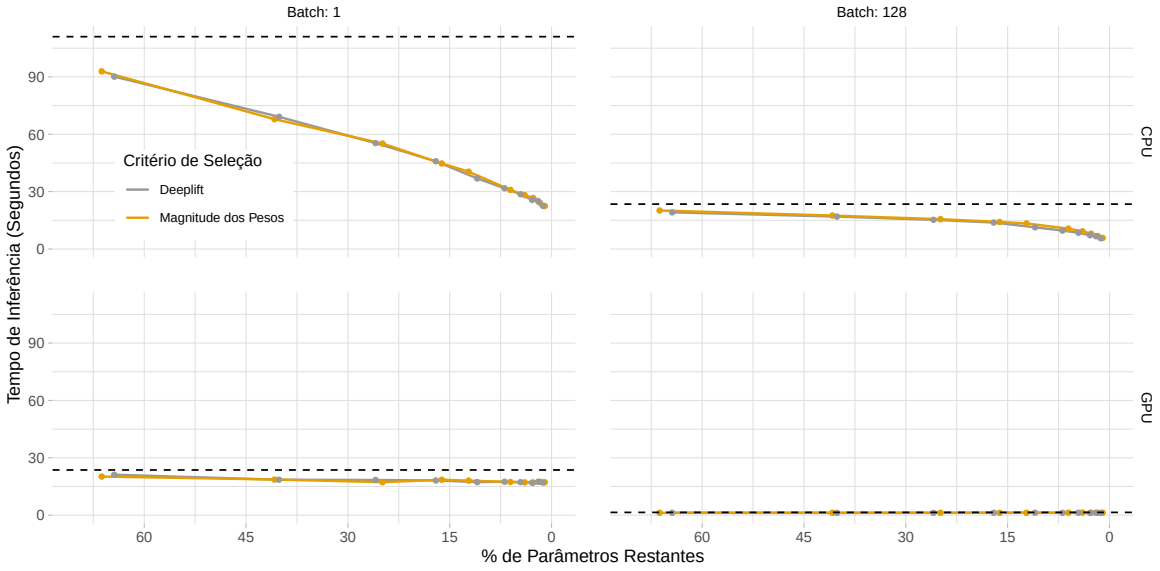


Figura 5.17: Tempo de inferência em segundos na partição de teste do conjunto de dados CIFAR-10 das redes podadas com magnitude dos pesos e DeepLIFT obtidas a partir da CNN VGG16. A linha tracejada representa o tempo de inferência da rede não podada.

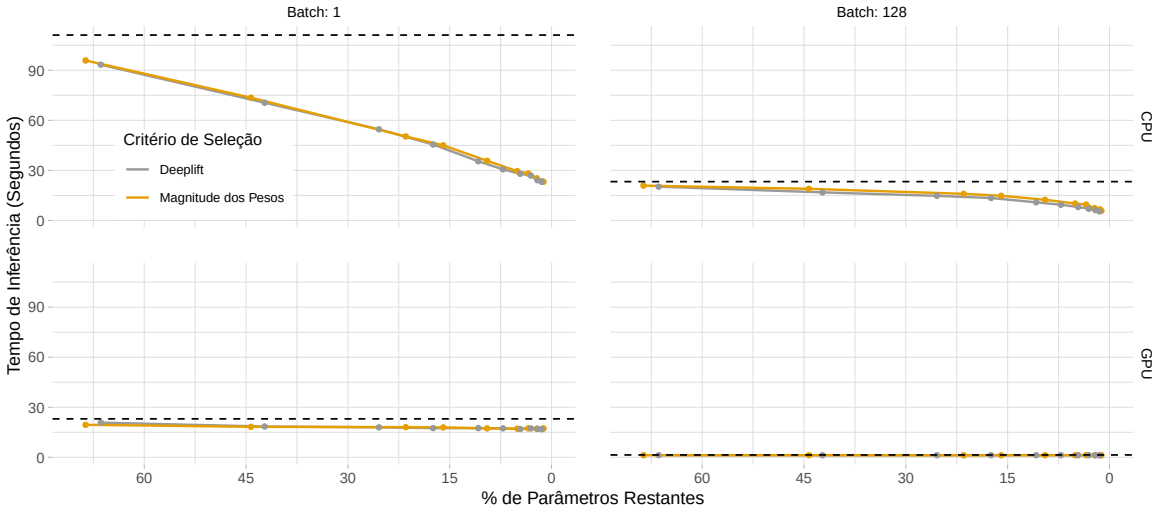


Figura 5.18: Tempo de inferência em segundos na partição de teste do conjunto de dados CIFAR-100 das redes podadas com magnitude dos pesos e DeepLIFT obtidas a partir da CNN VGG16. A linha tracejada representa o tempo de inferência da rede não podada.

Capítulo 6

Conclusões

Nesta pesquisa, avaliou-se experimentalmente as abordagens de retreinamento relacionadas à Hipótese do Bilhete de Loteria (FRANKLE; CARBIN, 2019; FRANKLE et al., 2019; RENDA; FRANKLE; CARBIN, 2020), Rebobinamento dos Pesos e Rebobinamento da Taxa de Aprendizagem, contribuindo assim com evidências empíricas que dão suporte à identificação de bilhetes vencedores ao realizar poda estruturada de redes neurais convolucionais. Além disso, propôs-se a utilização do método de explicabilidade DeepLIFT (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017) como alternativa à utilização da magnitude dos pesos como critério de seleção dos elementos a serem removidos durante o processo de poda, em conjunto com a abordagem de rebobinamento da taxa de aprendizagem, avaliando o impacto desta substituição nas redes neurais podadas.

Os resultados apresentados na Seção 5.1 evidenciam o surgimento de bilhetes vencedores quando aplicada poda global de forma estruturada. Poucos bilhetes vencedores puderam ser identificados ao aplicar a abordagem de Rebobinamento dos Pesos e as redes podadas com esta abordagem desempenharam igual ou pior que suas contrapartes treinadas com pesos iniciais aleatórios. Por outro lado, a abordagem de Rebobinamento da Taxa de Aprendizagem foi capaz de identificar bilhetes vencedores mesmo em níveis de poda agressivos e as redes podadas com esta abordagem desempenharam melhor que suas contrapartes treinadas com pesos iniciais aleatórios.

Os resultados apresentados na Seção 5.2 evidenciam que a substituição da magnitude dos pesos pelo método DeepLIFT melhora o desempenho das redes neurais convolucionais podadas, diminuindo a degradação da acurácia das redes podadas e tornando o processo de

poda mais estável, principalmente em níveis de poda mais agressivos. Essas melhorias têm relação com a distribuição por camada dos filtros removidos a cada iteração de poda, como sugere a análise apresentada na subseção 5.2.3.

Esses resultados, combinados à análise apresentada na subseção 5.2.4, dão suporte à viabilidade da produção de redes podadas mais a adequadas à execução em dispositivos com poucos recursos computacionais, uma vez que essas redes podadas apresentam diminuição real do tempo de execução, consumo de memória e armazenamento, sem degradação significativa da acurácia.

6.1 Trabalhos Futuros

Nesta seção são apresentadas possibilidades de trabalhos futuros a partir dos resultados desta pesquisa, com foco em aprofundar o entendimento dos métodos e validar a generalidade das conclusões, como descritos abaixo:

- Análise e identificação das características que fazem com que as abordagens de rebobinamento dos pesos e rebobinamento da taxa de aprendizagem produzam resultados diferentes em redes neurais convolucionais podadas de forma estruturada e não estruturada.

- Ampliar os experimentos desta pesquisa para mais conjuntos de dados (e.g. ImageNet (RUSSAKOVSKY et al., 2015) e Places365 (ZHOU, B. et al., 2017)), arquiteturas de redes neurais (por exemplo, ResNets (HE, K. et al., 2016) e DenseNet (HUANG et al., 2017)) e para diferentes tarefas (e.g., *data augmentation* dados usando Redes Adversárias Generativas (GOODFELLOW et al., 2014)).

- Aprofundar a compreensão sobre o método DeepLIFT enquanto critério de seleção de elementos a serem removidos durante o processo de poda, identificando cenários onde seu uso é mais indicado e também possíveis limitações. A visualização dos filtros convolucionais poderia ser uma ferramenta útil nesse contexto.

- Testar outras métricas de explicabilidade (e.g. *Conductance* (DHAMDHERE; SUNDARARAJAN; YAN, 2019), *Layer-Wise Relevance Propagation* (MONTAVON, Gré-

goire et al., 2019) e SHAP (LUNDBERG; LEE, 2017)) enquanto critério de poda de redes neurais profundas.

- Avaliar a relação entre a adoção de uma métrica de explicabilidade enquanto critério de poda e a explicabilidade da rede neural podada com base neste critério.

- Avaliar como a poda de redes neurais profundas se relaciona com o problema de *underspecification* (D'AMOUR et al., 2020).

- Avaliar o impacto das abordagens de poda de redes neurais convolucionais apresentadas nesta dissertação na vulnerabilidade das redes podadas a ataques adversariais [1].

[1] Redes Neurais Convolucionais estão sujeitas a ataques adversariais, que consistem na adição de ruído imperceptível a uma imagem de teste, fazendo com que a CNN classifique-a de forma errada (SZEGEDY et al., 2014).

Referências

BA, Jimmy; CARUANA, Rich. Do Deep Nets Really Need to be Deep? In: GHAHRAMANI, Z. et al. (Ed.). **Advances in Neural Information Processing Systems 27**. [S.l.]: Curran Associates, Inc., 2014. p. 2654–2662.

BACH, Sebastian et al. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. **PLOS ONE**, Public Library of Science, v. 10, n. 7, p. 1–46, jul. 2015. DOI: 10.1371/journal.pone.0130140. Disponível em: <https://doi.org/10.1371/journal.pone.0130140>.

CHEN, Jianda; CHEN, Shangyu; PAN, Sinno Jialin. Storage Efficient and Dynamic Flexible Runtime Channel Pruning via Deep Reinforcement Learning. In: _____. **Advances in Neural Information Processing Systems**. [S.l.]: Curran Associates, Inc., 2020. v. 33, p. 14747–14758. Disponível em: <https://proceedings.neurips.cc/paper/2020/file/a914ecef9c12ffdb9bede64bb703d877-Paper.pdf>.

CHOUDHARY, Tejalal et al. A comprehensive survey on model compression and acceleration. **Artificial Intelligence Review**, Springer, 2020. DOI: https://doi.org/10.1007/s10462-020-09816-7.

D'AMOUR, Alexander et al. Underspecification presents challenges for credibility in modern machine learning. **arXiv preprint arXiv:2011.03395**, 2020.

DHAMDHERE, Kedar; SUNDARARAJAN, Mukund; YAN, Qiqi. How Important is a Neuron? In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2019. Disponível em: <https://openreview.net/forum?id=SylKoo0cKm>.

DU, Simon S; LEE, Jason D. On the power of over-parametrization in neural networks with quadratic activation. **arXiv preprint arXiv:1803.01206**, 2018.

FRANKLE, Jonathan; CARBIN, Michael. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2019.

FRANKLE, Jonathan et al. Stabilizing the lottery ticket hypothesis. **arXiv preprint arXiv:1903.01611**, 2019.

GALE, Trevor; ELSEN, Erich; HOOKER, Sara. The State of Sparsity in Deep Neural Networks. **ArXiv**, abs/1902.09574, 2019.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. [S.l.]: MIT press, 2016.

GOODFELLOW, Ian et al. Generative adversarial nets. **Advances in neural information processing systems**, v. 27, 2014.

HAN, Song et al. Learning Both Weights and Connections for Efficient Neural Networks. In: PROCEEDINGS of the 28th International Conference on Neural Information Processing Systems - Volume 1. Montreal, Canada: MIT Press, 2015. (NIPS'15), p. 1135–1143.

HASSIBI, Babak; STORK, David G. Second order derivatives for network pruning: Optimal brain surgeon. In: ADVANCES in neural information processing systems. [S.l.: s.n.], 1993. p. 164–171.

HE, Kaiming et al. Deep residual learning for image recognition. In: PROCEEDINGS of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2016. p. 770–778.

HE, Yang et al. Filter Pruning via Geometric Median for Deep Convolutional Neural Networks Acceleration. In: THE IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], jun. 2019.

HE, Yang et al. Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks. In: PROCEEDINGS of the 27th International Joint Conference on Artificial Intelligence. Stockholm, Sweden: AAAI Press, 2018. (IJCAI'18), p. 2234–2240. ISBN 9780999241127.

HE, Yihui; ZHANG, Xiangyu; SUN, Jian. Channel pruning for accelerating very deep neural networks. In: PROCEEDINGS of the IEEE International Conference on Computer Vision. [S.l.: s.n.], 2017. p. 1389–1397.

HENNESSY, John L.; PATTERSON, David A. **Computer Architecture, Sixth Edition: A Quantitative Approach**. 6th. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2017. ISBN 0128119055.

HUANG, Gao et al. Densely connected convolutional networks. In: PROCEEDINGS of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2017. p. 4700–4708.

IOFFE, Sergey; SZEGEDY, Christian. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: INTERNATIONAL Conference on Machine Learning. [S.l.: s.n.], 2015. p. 448–456.

JAIN, Anil K.; JIANCHANG MAO; MOHIUDDIN, K. M. Artificial neural networks: a tutorial. **Computer**, v. 29, n. 3, p. 31–44, mar. 1996. ISSN 1558-0814. DOI: 10.1109/2.485891.

KINGMA, Diederik; BA, Jimmy. Adam: A Method for Stochastic Optimization. **International Conference on Learning Representations**, dez. 2014.

KOKHLIKYAN, Narine et al. **Captum: A unified and generic model interpretability library for PyTorch**. [S.l.: s.n.], 2020. arXiv: 2009.07896 [cs.LG].

KRIZHEVSKY, Alex. **Learning multiple layers of features from tiny images**. 2009. Diss. (Mestrado) – University of Toronto.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.

LECUN, Yann; DENKER, John S; SOLLA, Sara A. Optimal brain damage. In: ADVANCES in neural information processing systems. [S.l.: s.n.], 1990. p. 598–605.

LECUN, Yann; KAVUKCUOGLU, Koray; FARABET, Clément. Convolutional networks and applications in vision. In: IEEE. PROCEEDINGS of 2010 IEEE international symposium on circuits and systems. [S.l.: s.n.], 2010. p. 253–256.

LECUN, Yann et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998. DOI: 10.1109/5.726791.

LI, Hao et al. Pruning Filters for Efficient ConvNets. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2017.

LI, Yuanzhi; LIANG, Yingyu. Learning Overparameterized Neural Networks via Stochastic Gradient Descent on Structured Data. In: BENGIO, S. et al. (Ed.). **Advances in Neural Information Processing Systems 31**. [S.l.]: Curran Associates, Inc., 2018. p. 8157–8166.

LIU, Weibo et al. A survey of deep neural network architectures and their applications. **Neurocomputing**, Elsevier, v. 234, p. 11–26, 2017.

LIU, Zechun et al. MetaPruning: Meta Learning for Automatic Neural Network Channel Pruning. In: PROCEEDINGS of the IEEE/CVF International Conference on Computer Vision (ICCV). [S.l.: s.n.], out. 2019.

LIU, Zhuang et al. Rethinking the Value of Network Pruning. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2019.

LIVNI, Roi; SHALEV-SHWARTZ, Shai; SHAMIR, Ohad. On the computational efficiency of training neural networks. In: ADVANCES in neural information processing systems. [S.l.: s.n.], 2014. p. 855–863.

LUNDBERG, Scott M; LEE, Su-In. A unified approach to interpreting model predictions. In: PROCEEDINGS of the 31st international conference on neural information processing systems. [S.l.: s.n.], 2017. p. 4768–4777.

LUO, Jian-Hao; WU, Jianxin; LIN, Weiyao. Thinet: A filter level pruning method for deep neural network compression. In: PROCEEDINGS of the IEEE international conference on computer vision. [S.l.: s.n.], 2017. p. 5058–5066.

MAGALHÃES, Whendell F. et al. Evaluating the Emergence of Winning Tickets by Structured Pruning of Convolutional Networks. In: 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). [S.l.: s.n.], 2020. p. 272–279. DOI: 10.1109/SIBGRAPI51738.2020.00044.

MILLER, Tim. Explanation in artificial intelligence: Insights from the social sciences. **Artificial Intelligence**, v. 267, p. 1–38, 2019. ISSN 0004-3702. DOI: https://doi.org/10.1016/j.artint.2018.07.007. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0004370218305988>.

MOLNAR, Christoph. **Interpretable Machine Learning**: A Guide for Making Black Box Models Explainable. [S.l.: s.n.], 2019.

MONTAVON, Grégoire et al. Explaining nonlinear classification decisions with deep Taylor decomposition. **Pattern Recognition**, Elsevier BV, v. 65, p. 211–222, mai. 2017. ISSN 0031-3203. DOI: 10.1016/j.patcog.2016.11.008. Disponível em: <http://dx.doi.org/10.1016/j.patcog.2016.11.008>.

MONTAVON, Grégoire et al. Layer-Wise Relevance Propagation: An Overview. In: **Explainable AI: Interpreting, Explaining and Visualizing Deep Learning**. Edição: Wojciech Samek. Cham: Springer International Publishing, 2019. p. 193–209. ISBN 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_10. Disponível em: <https://doi.org/10.1007/978-3-030-28954-6_10>.

MORCOS, Ari et al. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. In: ADVANCES in Neural Information Processing Systems. [S.l.: s.n.], 2019. p. 4932–4942.

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

NWANKPA, Chigozie et al. Activation functions: Comparison of trends in practice and research for deep learning. **arXiv preprint arXiv:1811.03378**, 2018.

PASZKE, Adam et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: WALLACH, H. et al. (Ed.). **Advances in Neural Information Processing Systems 32**. [S.l.]: Curran Associates, Inc., 2019. p. 8024–8035. Disponível em: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.

PENG, Hanyu et al. Collaborative Channel Pruning for Deep Networks. In: _____. **Proceedings of the 36th International Conference on Machine Learning**. [S.l.]: PMLR, jun. 2019. v. 97. (Proceedings of Machine Learning Research), p. 5113–5122. Disponível em: <https://proceedings.mlr.press/v97/peng19c.html>.

QASSIM, Hussam; VERMA, Abhishek; FEINZIMER, David. Compressed residual-VGG16 CNN model for big data places image recognition. In: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). [S.l.: s.n.], 2018. p. 169–175. DOI: 10.1109/CCWC.2018.8301729.

RENDA, Alex; FRANKLE, Jonathan; CARBIN, Michael. Comparing Rewinding and Fine-tuning in Neural Network Pruning. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2020.

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.

SABIH, Muhammad; HANNIG, Frank; TEICH, Jürgen. Utilizing Explainable AI for Quantization and Pruning of Deep Neural Networks. **CoRR**, abs/2008.09072, 2020. arXiv: 2008.09072. Disponível em: <https://arxiv.org/abs/2008.09072>.

SALAMA, Abdullah et al. **Pruning at a Glance: Global Neural Pruning for Model Compression**. [S.l.: s.n.], 2019. arXiv: 1912.00200 [cs.CV].

SELVARAJU, Ramprasaath R. et al. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. **CoRR**, abs/1610.02391, 2016. arXiv: 1610.02391. Disponível em: <http://arxiv.org/abs/1610.02391>.

SHRIKUMAR, Avanti; GREENSIDE, Peyton; KUNDAJE, Anshul. Learning important features through propagating activation differences. In: PMLR. INTERNATIONAL Conference on Machine Learning. [S.l.: s.n.], 2017. p. 3145–3153.

SIMONYAN, Karen; VEDALDI, Andrea; ZISSERMAN, Andrew. Deep inside convolutional networks: Visualising image classification models and saliency maps. **arXiv preprint arXiv:1312.6034**, 2013.

SIMONYAN, Karen; ZISSERMAN, Andrew. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2015.

SMILKOV, Daniel et al. SmoothGrad: removing noise by adding noise. **CoRR**, abs/1706.03825, 2017. arXiv: 1706.03825. Disponível em: <http://arxiv.org/abs/1706.03825>.

SZEGEDY, Christian et al. Intriguing properties of neural networks. In: _____. **2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings**. [S.l.: s.n.], 2014. Disponível em: <http://arxiv.org/abs/1312.6199>.

VAN ROSSUM, Guido; DRAKE, Fred L. **Python 3 Reference Manual**. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.

WANG, Zi; LI, Chengcheng. Channel Pruning via Lookahead Search Guided Reinforcement Learning. In: PROCEEDINGS of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). [S.l.: s.n.], jan. 2022. p. 2029–2040.

WU, Huibo et al. An Efficient Pruning Method of Digital Predistortion Suitable for Power Amplifiers with Scalable Output Power. In: 2021 IEEE MTT-S International Wireless Symposium (IWS). [S.l.: s.n.], 2021. p. 1–3. DOI: 10.1109/IWS52775.2021.9499681.

WU, Yonghui et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. **CoRR**, abs/1609.08144, 2016. arXiv: 1609.08144. Disponível em: <http://arxiv.org/abs/1609.08144>.

YEOM, Seul-Ki et al. Pruning by explaining: A novel criterion for deep neural network pruning. **Pattern Recognition**, v. 115, p. 107899, 2021. ISSN 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2021.107899. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0031320321000868>.

ZHANG, Zhengyan et al. Know what you don't need: Single-Shot Meta-Pruning for attention heads. **AI Open**, v. 2, p. 36–42, 2021. ISSN 2666-6510. DOI: https://doi.org/10.1016/j.aiopen.2021.05.003. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666651021000140>.

ZHOU, Bolei et al. Places: A 10 million Image Database for Scene Recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, 2017.

ZHOU, Hattie et al. Deconstructing lottery tickets: Zeros, signs, and the supermask. In: ADVANCES in Neural Information Processing Systems. [S.l.: s.n.], 2019. p. 3597–3607.

Apêndice A

Filtros Removidos por Iteração de Poda

As figuras deste apêndice exibem a evolução da distribuição dos filtros convolucionais podados a cada iteração de poda na rede neural convolucional VGG16, treinada nos conjuntos de dados CIFAR-10 e CIFAR-100, podada de forma global, retreinada usando rebobinamento dos pesos e os critérios de poda baseados na Magnitude dos Pesos e no método DeepLIFT.

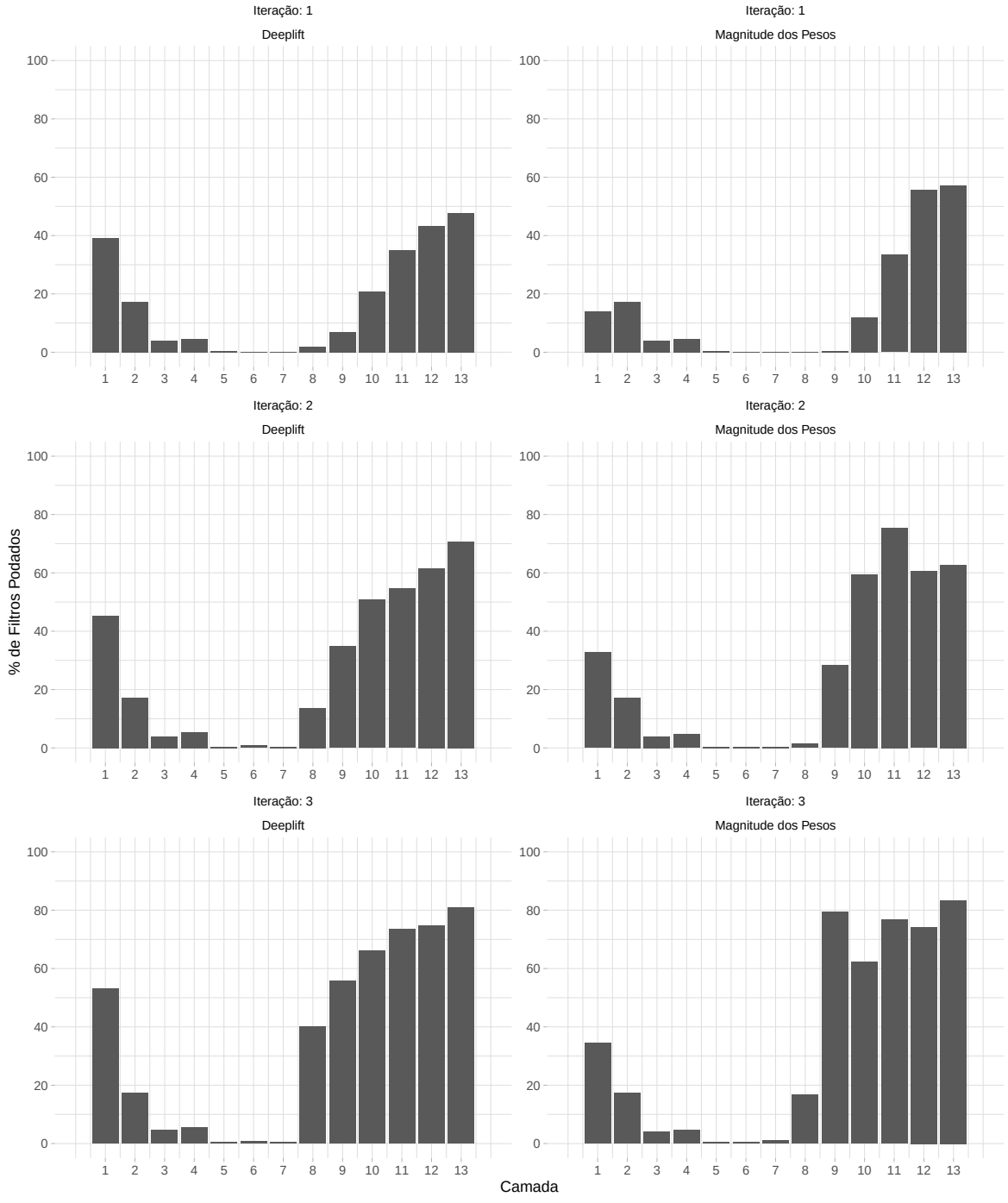


Figura A.1: Percentual de filtros convolucionais podados com DeepLIFT e com Magnitude dos Pesos nas iterações de poda 1, 2 e 3 da rede neural convolucional VGG16 no conjunto de dados CIFAR-10.

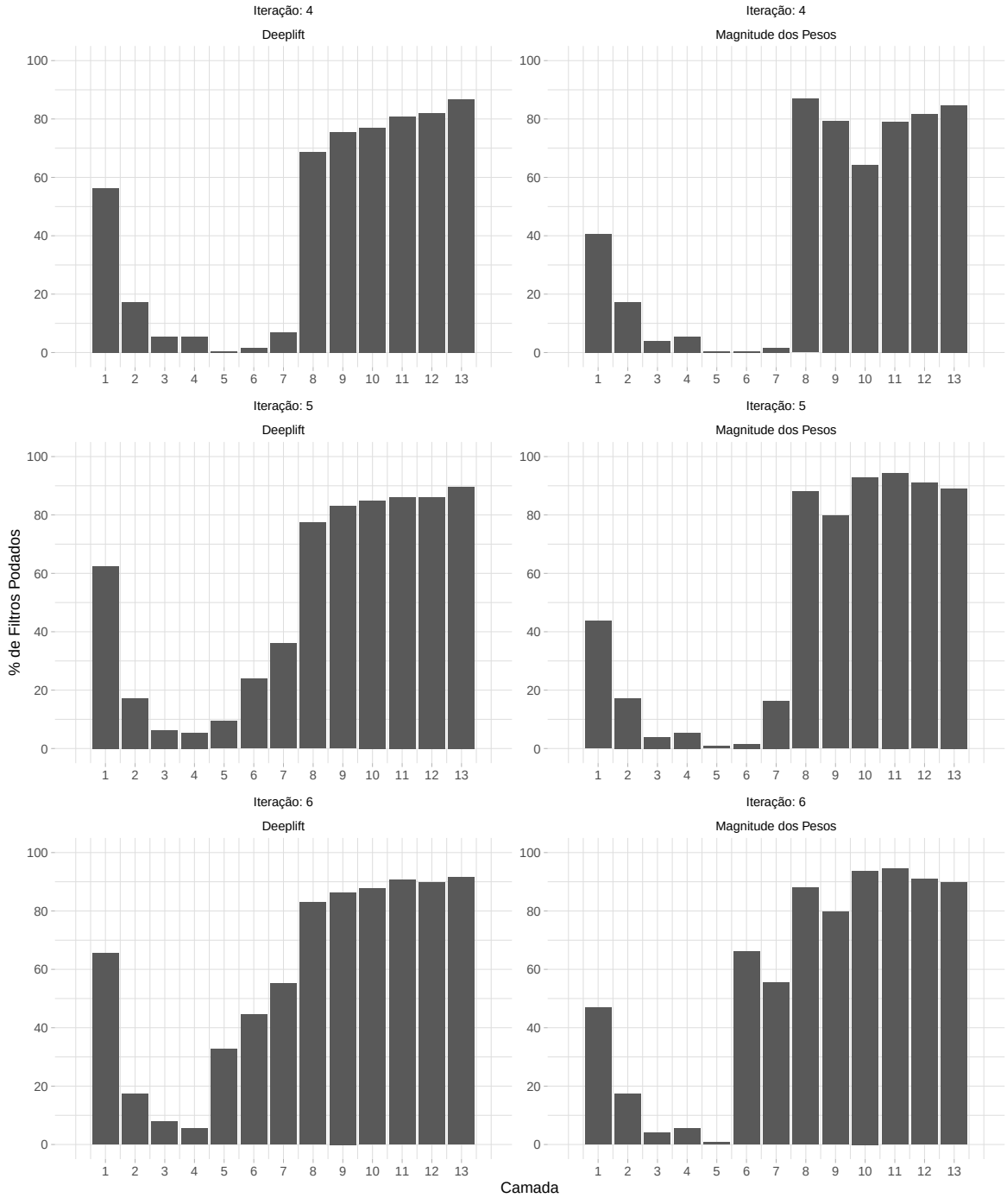


Figura A.2: Percentual de filtros convolucionais podados com DeepLIFT e com Magnitude dos Pesos nas iterações de poda 4, 5 e 6 da rede neural convolucional VGG16 no conjunto de dados CIFAR-10.

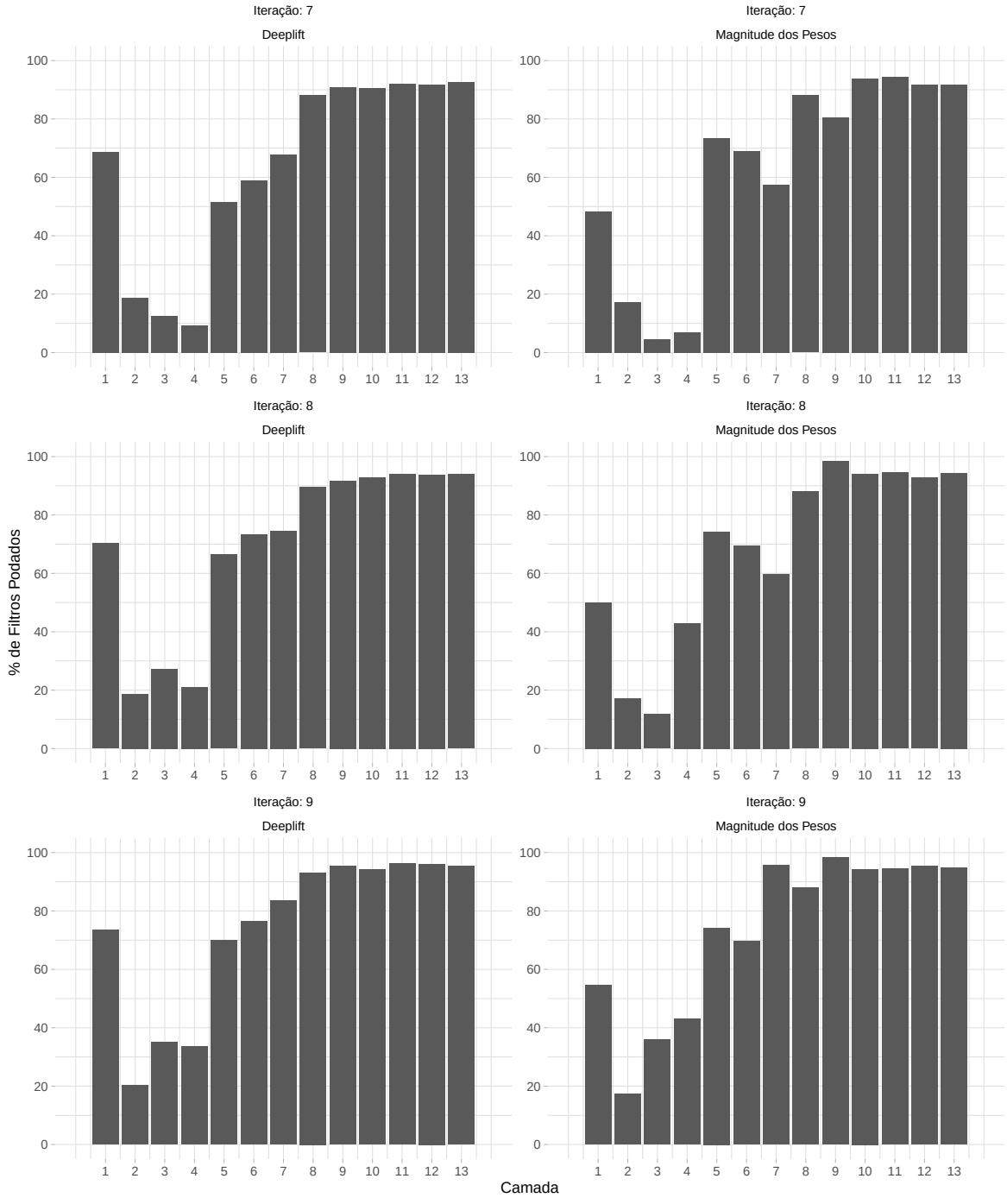


Figura A.3: Percentual de filtros convolucionais podados com DeepLIFT e com Magnitude dos Pesos nas iterações de poda 7, 8 e 9 da rede neural convolucional VGG16 no conjunto de dados CIFAR-10.

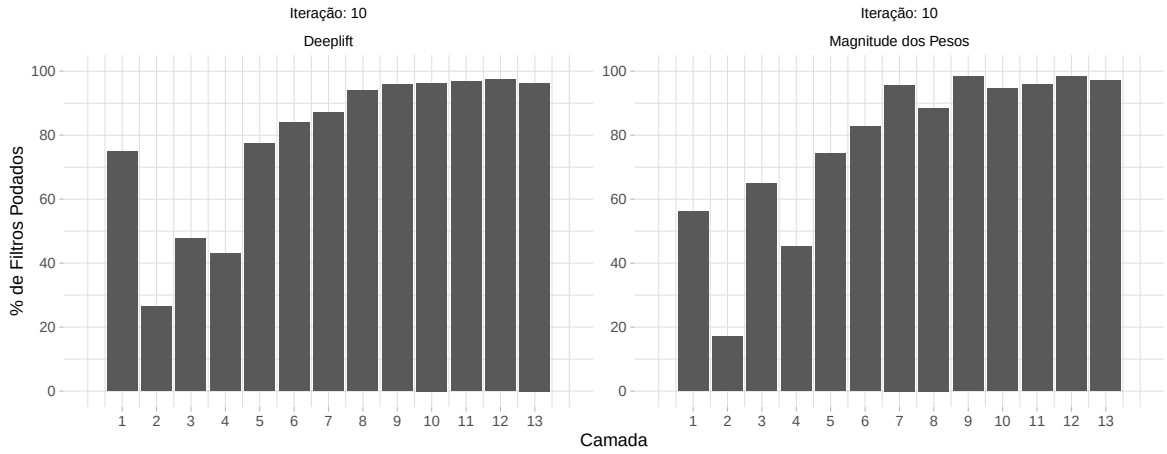


Figura A.4: Percentual de filtros convolucionais podados com DeepLIFT e com Magnitude dos Pesos na iteração de poda 10 da rede neural convolucional VGG16 no conjunto de dados CIFAR-10.

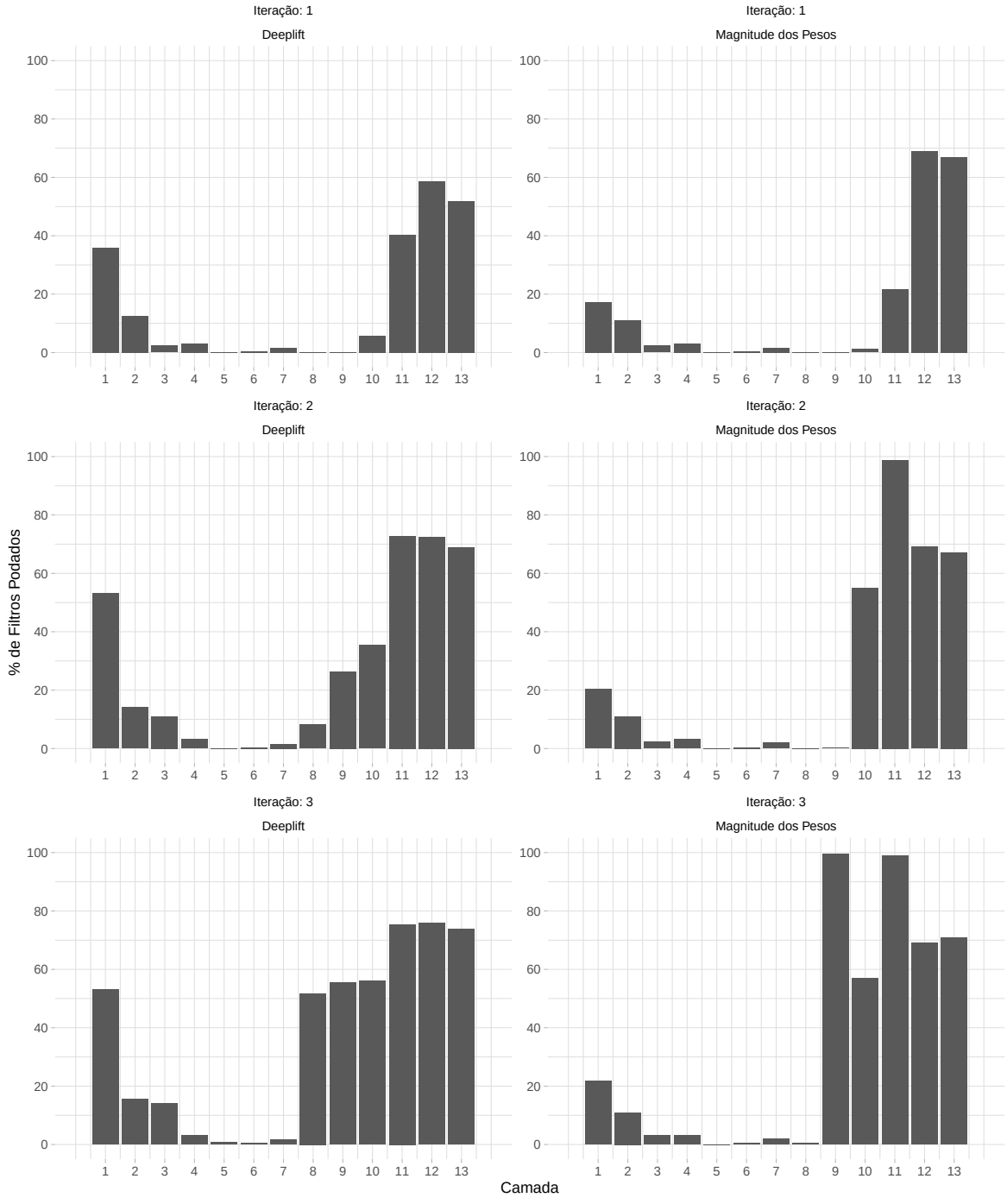


Figura A.5: Percentual de filtros convolucionais podados com DeepLIFT e com Magnitude dos Pesos nas iterações de poda 1, 2 e 3 da rede neural convolucional VGG16 no conjunto de dados CIFAR-100.

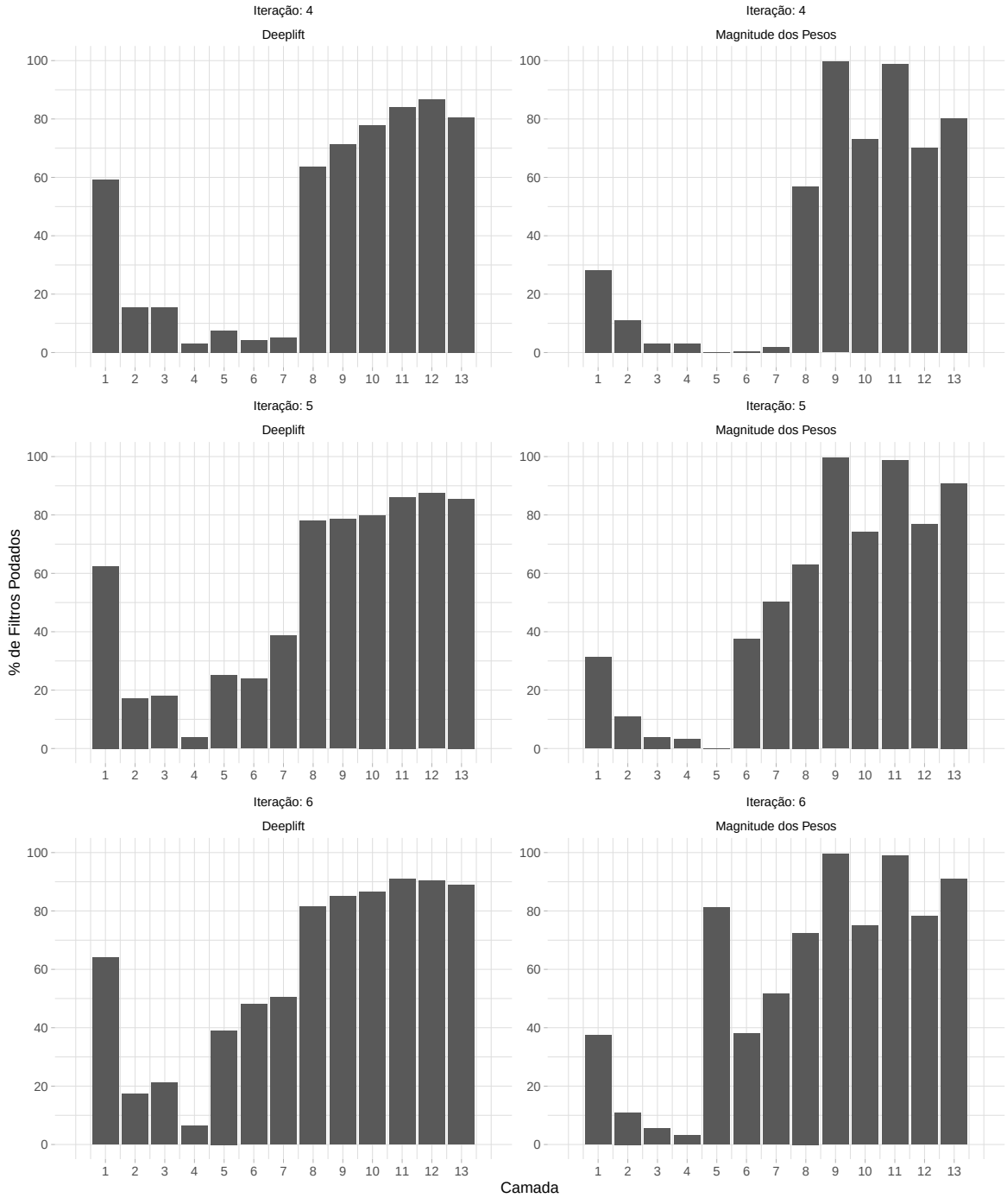


Figura A.6: Percentual de filtros convolucionais podados com DeepLIFT e com Magnitude dos Pesos nas iterações de poda 4, 5 e 6 da rede neural convolucional VGG16 no conjunto de dados CIFAR-100.

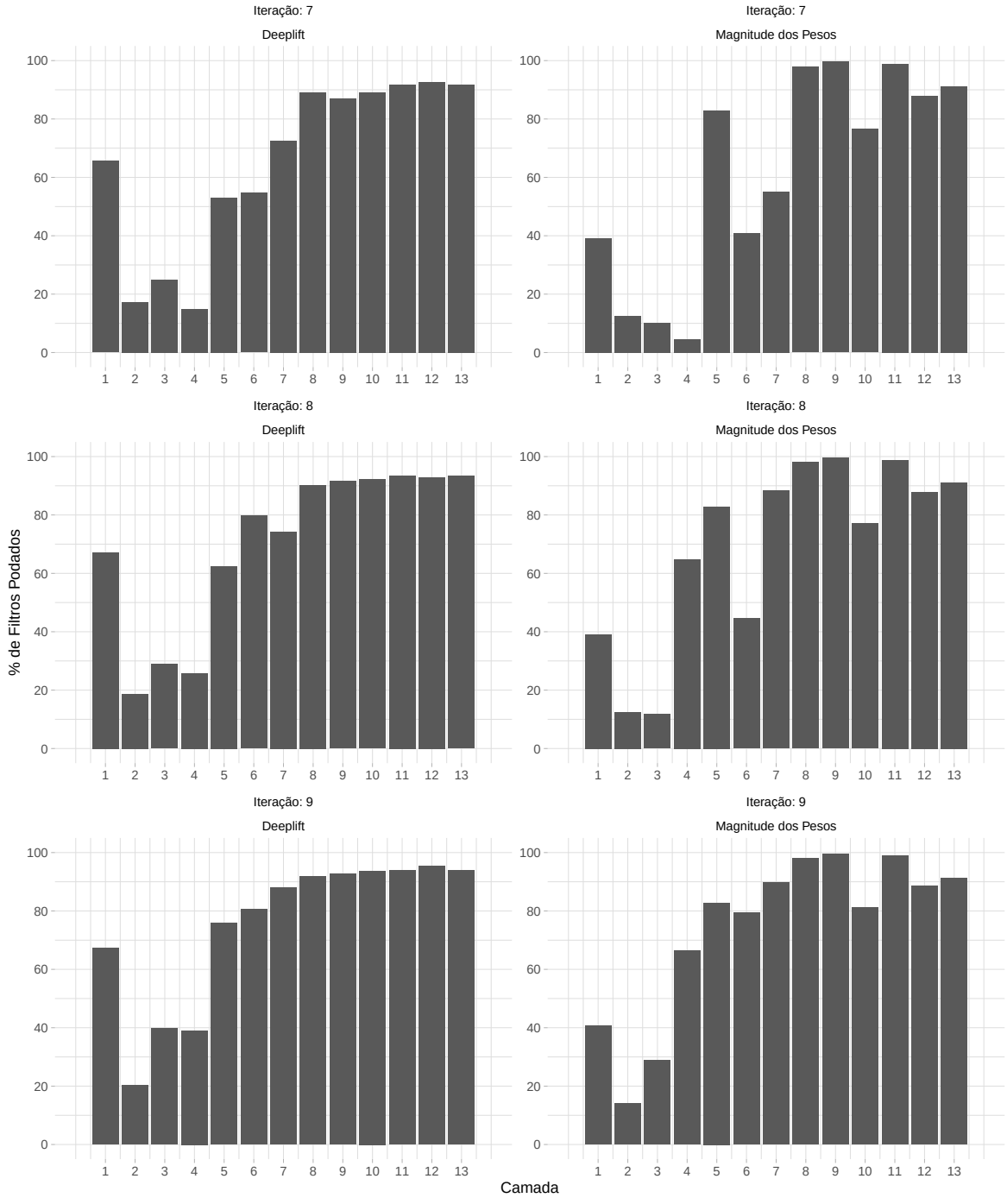


Figura A.7: Percentual de filtros convolucionais podados com DeepLIFT e com Magnitude dos Pesos nas iterações de poda 7, 8 e 9 da rede neural convolucional VGG16 no conjunto de dados CIFAR-100.

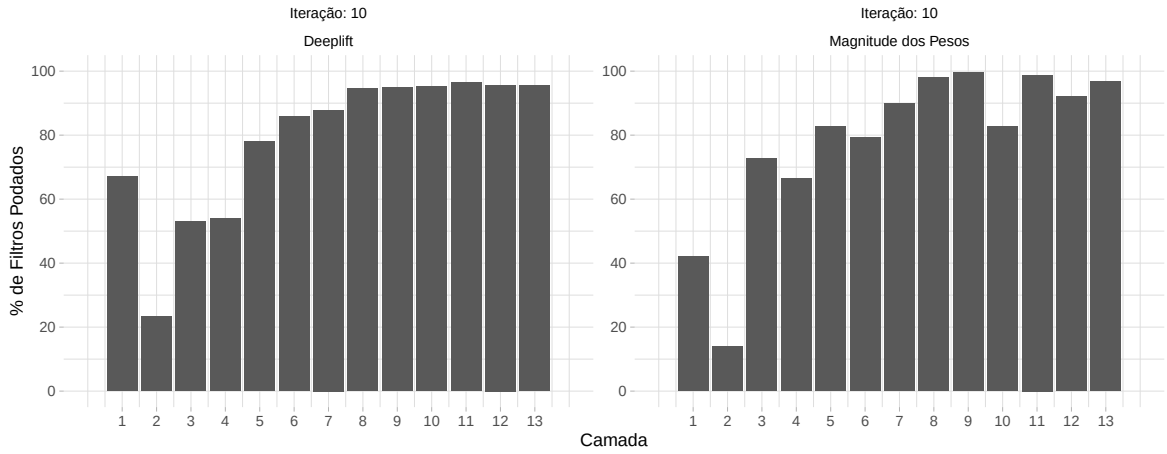


Figura A.8: Percentual de filtros convolucionais podados com DeepLIFT e com Magnitude dos Pesos na iteração de poda 10 da rede neural convolucional VGG16 no conjunto de dados CIFAR-100.