CARACTERIZAÇÃO DO DESEMPENHO DE CNN'S EM DISPOSITIVOS COM DIFERENTES CAPACIDADES COMPUTACIONAIS

GUSTAVO HENRIQUE ROOS[1], ULISSES BRISOLARA CORRÊA[2], PAULO ROBERTO FERREIRA JÚNIOR[3]

[1] *Universidade Federal de Pelotas – ghroos@inf.ufpel.edu.br*
[2] *Universidade Federal de Pelotas – ub.correa@inf.ufpel.edu.br*
[3] *Universidade Federal de Pelotas – paulo.ferreira@inf.ufpel.edu.br*

1. INTRODUÇÃO

As Redes Neurais Convolucionais (CNNs) são uma variante das Redes Neurais Artificiais Profundas (DNNs) que se destacam em aplicações que envolvem processamento de imagens, visão computacional e processamento de linguagem natural (NLP). Ela é composta por várias camadas, incluindo convolucionais, de não-linearidade e pooling. As CNNs demonstram um desempenho excepcional em problemas de aprendizado de máquina, principalmente em tarefas que lidam com dados de imagem, como classificação de imagens, e seus resultados têm sido notáveis (ALBAWI; MOHAMMED; AL-ZAWI, 2017). Em termos simples, as CNNs são redes neurais que podem ser usadas para extrair características de imagens, como bordas, texturas e formas, e posteriormente utilizar essas características para classificar ou identificar objetos contidos nas imagens.

O tempo de inferência das CNNs é influenciado por diversos fatores, que incluem o tamanho do modelo, a arquitetura da rede neural e o *hardware* em que a rede está sendo executada. Este trabalho se propõe a realizar uma comparação do desempenho de 10 CNNs em dois dispositivos distintos: uma Nvidia Jetson Nano e uma Nvidia RTX 3060.

Os principais objetivos deste trabalho são:

- comparar o desempenho de diferentes CNNs em contextos distintos, utilizando dispositivos diversos;
- investigar como o tamanho do modelo e a arquitetura da rede neural impactam o desempenho nos dispositivos utilizados;
- fornecer recomendações práticas para a utilização de CNNs em tarefas de classificação de imagens em diferentes dispositivos.

2. METODOLOGIA

Para conduzir este estudo, selecionamos um conjunto de dados composto por 95 imagens escolhidas aleatoriamente a partir da partição de validação do conjunto de dados ImageNet 1K. Esse conjunto é composto por milhões de imagens categorizadas em 1000 classes distintas (RUSSAKOVSKY et al., 2015).

As inferências foram realizadas utilizando diferentes tamanhos de lote, especificamente 1, 4, 8, 16 e 32 imagens por iteração. O tamanho do lote é o número de imagens que são processadas em paralelo pela rede neural. Um tamanho de lote

Agradecemos à CNPq, à FAPERGS e à NVIDIA Corporation pelo financiamento parcial deste trabalho.

maior pode melhorar a eficiência da inferência, mas também pode aumentar o uso de memória e processamento.

Os seguintes modelos foram selecionados para a execução das inferências: ResNets 18, 34, 50, 101 e 152 (HE et al., 2015), MobileNet V2 (SANDLER et al., 2019), MobileNet V3 Small (HOWARD et al., 2019), MobileNet V3 Large (HOWARD et al., 2019), Inception V3 (SZEGEDY et al., 2015) e GoogLeNet (SZEGEDY et al., 2014). Todos os modelos foram utilizados com pesos pré-treinados na partição de treino do conjunto de dados ImageNet 1K. Isso foi possível através do framework PyTorch, o qual fornece uma biblioteca de funções para carregar e usar modelos com pesos pré-treinados.

Para coletar os dados de pico de memória, foi utilizada a biblioteca *memory-profiler*, utilizada para calcular o pico máximo de memória consumida por um código Python (PEDREGOSA, 2022).

Para coletar os dados de consumo médio em watts na Nvidia Jetson Nano foi realizado o monitoramento de um arquivo do sistema operacional que possui o valor de entrada de potência atual. Foi realizada a coleta destes valores a cada 500 milissegundos durante a execução do processo e subsequentemente um cálculo da média destes valores para estimar o consumo energético. Na placa de vídeo Nvidia RTX 3060 foi realizada a utilização da biblioteca *pynvml*, a qual fornece uma interface Python para funções de gerenciamento e monitoramento de GPUs (NVIDIA, 2023). Com esta biblioteca, foi possível realizar a coleta do consumo da GPU a cada 500 milissegundos, possibilitando a realização do cálculo da mesma forma que fora realizado para a Nvidia Jetson Nano.

Na análise dos resultados, consideramos os seguintes indicadores: acurácia, tempo de inferência, tamanho do lote, pico máximo de memória e consumo de energia médio.

3. RESULTADOS E DISCUSSÃO

Para a coleta dos resultados, cada configuração de tamanho de lote e rede foi executada 100 vezes. Posteriormente, as médias dos valores foram calculadas, com o objetivo de mitigar eventuais variações decorrentes do ambiente físico em que cada dispositivo estava inserido. Isso incluiu fatores como a temperatura ambiente, a temperatura do dispositivo e a execução de processos concorrentes no sistema operacional. Os resultados obtidos para a placa de vídeo Nvidia RTX 3060 e a Jetson Nano estão detalhados nas Tabelas 1 e 2, respectivamente.

Além das tabelas, a partir dos experimentos, foi possível observar um notável aumento de desempenho à medida que o tamanho do lote foi incrementado em ambos os dispositivos. Esse aumento se deve ao fato de que um lote é processado de forma paralela na GPU, possibilitando um processamento mais rápido, aproveitando ao máximo os recursos de hardware disponíveis.

Assim como a placa de vídeo RTX 3060, que dispõe de uma quantidade substancialmente maior de memória, observou-se que a Jetson Nano teve seu tempo médio de inferência reduzido até o lote de tamanho 16. No entanto, ao contrário da RTX 3060, ao aumentar o tamanho do lote de 16 para 32 imagens, o tempo médio de inferência na Jetson Nano apresentou um aumento significativo.

MODELO	ACURÁCIA	TEMPO MÉDIO (s)	PICO MÉDIO (MB)	CONSUMO MÉDIO (W)
ResNet18	0,7158	0,8868	2918,6394	31,1068
ResNet34	0,7684	0,9494	2921,4142	31,9260
ResNet50	0,8000	1,0581	2929,3802	33,4398
ResNet101	0,7684	1,2486	2945,2262	34,8336
ResNet152	0,8105	1,4427	2963,7086	36,4935
MobileNet V2	0,7474	0,9646	2910,9732	31,4582
MobileNet V3 Small	0,7263	0,9782	2915,1690	30,5052
MobileNet V3 Large	0,7474	1,0123	2917,3973	30,6362
GoogLeNet	0,7895	1,0920	2933,7257	31,8344
Inception V3	0,7579	1,2308	2944,3495	35,6234

Tabela 1: Inferências realizadas na RTX 3060.

MODELO	ACURÁCIA	TEMPO MÉDIO (s)	PICO MÉDIO (MB)	CONSUMO MÉDIO (W)
ResNet18	0,7158	11,5878	2522,0749	3,5263
ResNet34	0,7684	14,8394	2468,7842	3,5952
ResNet50	0,8000	19,7057	2415,5143	3,7559
ResNet101	0,7684	25,5194	2366,5987	3,9772
ResNet152	0,8105	30,2095	2347,4491	4,2046
MobileNet V2	0,7579	12,1007	2519,4996	3,5522
MobileNet V3 Small	0,7263	7,6355	2600,8271	3,4615
MobileNet V3 Large	0,7474	9,1528	2550,1302	3,5342
GoogLeNet	0,7895	12,1923	2495,0400	3,7082
Inception V3	0,7579	29,6407	2320,8450	3,8481

Tabela 2: Inferências realizadas na Jetson Nano.

Esse comportamento é atribuído à limitação de memória presente na Jetson Nano em comparação com a RTX 3060, o que dificulta a manipulação de lotes maiores, resultando em sobrecarga da memória e, consequentemente, em tempos de inferência mais longos. Isso destaca a importância de considerar a capacidade de memória de hardware ao escolher o tamanho do lote em aplicações com redes neurais convolucionais, especialmente em dispositivos com recursos limitados, como a Jetson Nano.

4. CONCLUSÃO

Com base nos resultados experimentais apresentados neste trabalho, podemos fornecer as seguintes recomendações práticas para a utilização de CNNs em tarefas de classificação de imagens:

- para aplicações em tempo real em dispositivos com capacidade limitada, como a Jetson Nano, pode ser necessário: abrir mão da acurácia para que a tarefa possa ser realizada em tempo hábil; utilizar uma arquitetura mais sim-

ples, a qual utiliza menos memória e faz um consumo menor de energia, como por exemplo as arquiteturas MobileNet V2, MobileNet V3 Small, MobileNet V3 Large, ResNet 18 ou ResNet 34. Outra sugestão seria utilizar técnicas de otimização para reduzir o uso de memória e energia, como a quantização de redes neurais (NAGEL et al., 2021);

- para aplicações que não demandam resposta em tempo real, i.e., que realizam um processamento por dia, por exemplo, é possível utilizar arquiteturas mais complexas, como a ResNet 152, para garantir uma taxa de acurácia mais alta tendo em vista que o consumo de energia não terá um impacto tão grande quanto em uma aplicação em tempo real.

Em resumo, a escolha da arquitetura da CNN, do tamanho do lote e do dispositivo deve ser feita com base nas necessidades específicas da aplicação, considerando fatores como custo, desempenho e impacto na experiência do usuário.

4. Referências

ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: **2017 International Conference on Engineering and Technology (ICET)**. [S.l.: s.n.], 2017. p. 1–6.

HE, K.; ZHANG, X.; REN, S.; SUN, J. **Deep Residual Learning for Image Recognition**. 2015.

HOWARD, A.; SANDLER, M.; CHU, G.; CHEN, L.-C.; CHEN, B.; TAN, M.; WANG, W.; ZHU, Y.; PANG, R.; VASUDEVAN, V.; LE, Q. V.; ADAM, H. **Searching for MobileNetV3**. 2019.

NAGEL, M.; FOURNARAKIS, M.; AMJAD, R. A.; BONDARENKO, Y.; BAALEN, M. van; BLANKEVOORT, T. **A White Paper on Neural Network Quantization**. 2021.

NVIDIA. **pynvml**. 2023. <https://pypi.org/project/pynvml/>. [Accessado em 22-09-2023].

PEDREGOSA, F. **memory-profiler**. 2022. <https://pypi.org/project/memory-profiler/>. [Accessado em 22-09-2023].

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; FEI-FEI, L. **ImageNet Large Scale Visual Recognition Challenge**. 2015.

SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; CHEN, L.-C. **MobileNetV2: Inverted Residuals and Linear Bottlenecks**. 2019.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. **Going Deeper with Convolutions**. 2014.

SZEGEDY, C.; VANHOUCKE, V.; IOFFE, S.; SHLENS, J.; WOJNA, Z. **Rethinking the Inception Architecture for Computer Vision**. 2015.