Avaliação do Consumo de Energia e o Impacto da emissão de CO_2 para algoritmos de Inteligência Artificial

Felipe Bernardo, Gabrieli Silva, Matheus Gritz, Mariza Ferro, Bruno Schulze

[1]Laboratório Nacional de Computação Científica (LNCC)
Getúlio Vargas, 333, Quitandinha – Petrópolis – Rio de Janeiro

{felipeb,gabrieli,masgritz,mariza,schulze}@lncc.br

Abstract. *Currently, Artificial Intelligence (AI) is one of the most transforming forces, achieving surprising results. These results are due, at large, to the use of high computational capacity offered by HPC environments, which at the same time require a lot of energy to keep them running. Energy consumption is responsible for greenhouse gas emissions, among which CO_2 is the most expressive. In this work, the impact of training different AI algorithms on energy consumption and equivalent CO_2 emissions for different computational architectures (ARM, GPU, and X86) is evaluated.*

Resumo. *Atualmente a Inteligência Artificial (IA) é uma das forças mais transformadoras do nosso tempo, com resultados surpreendentes. Esses resultados se devem, em grande parte, ao uso de alta capacidade computacional oferecida pelos ambientes de HPC, os quais ao mesmo tempo requerem muita energia para seu funcionamento. Além disso, o consumo de energia é responsável pela emissão de gases de efeito estufa, entre os quais o CO_2 é o mais expressivo. Neste trabalho é avaliado o impacto do treinamento de diferentes algoritmos de IA no consumo energético e na emissão de CO_2 equivalente entre diferentes arquiteturas computacionais (ARM, GPU e X86).*

1. Introdução

Na última década se tem debatido muito a importância da preservação da natureza. Um dos pontos que tange esse debate é o consumo energético, pois em muitos países, tais como China, EUA e Alemanha, a maioria das matrizes energéticas não utilizam fontes renováveis na geração de energia. Uma das áreas envolvidas diretamente com isso é a computação, já que os ambientes de Computação de Alto Desempenho (HPC[1]) consomem muita energia para manter o seu funcionamento, incluindo núcleos de processamento e sistemas de refrigeração. Neste contexto, com o propósito de alertar sobre o aumento significativo do consumo de energia elétrica desses ambientes, criou-se a lista Green500[2], uma classificação para os supercomputadores mais eficientes energeticamente. Além disso, existem esforços para mudar este cenário, utilizando fontes energeticamente renováveis para o abastecimento dos ambientes de HPC [Google 2019]. Outra iniciativa é a *Power Usage Effectiveness* (PUE) [Avelar et al. 2012], uma métrica para medir a eficiência energética dos ambientes de computação, o que facilita a obtenção de informações sobre o consumo de energia desses equipamentos e a sua otimização.

Por outro lado, esses ambientes de HPC se tornaram cruciais para a descoberta de novos conhecimentos em diversas áreas da ciência e engenharia. Entre elas estão as técnicas de IA e sua sub-área de Aprendizado de Máquina (AM). Entretanto, a execução de algoritmos de IA, principalmente no treinamento de algoritmos de AM e Redes Neurais Artificiais (RNAs), requerem grande quantidade de dados e capacidade computacional, por isso, a IA vem se beneficiando da HPC na solução de problemas nos mais diversos domínios de aplicação. Porém, mesmo

[1]HPC - Do termo em inglês High Performance Computing
[2]https://www.top500.org/green500/

com a relevância dos algoritmos de AM, pouco se sabe a respeito dos seus requisitos computacionais e consumo de energia em diferentes arquiteturas computacionais. O que se encontra são pesquisas comparativas entre a acurácia dos algoritmos para resolver um determinado problema [Malakar et al. 2018] ou para avaliação de algoritmos de RNAs em GPUs [Santos 2015].

Portanto, avaliar o consumo de energia dos algoritmos de IA em diferentes arquiteturas computacionais tornou-se tarefa necessária e relevante, sendo um dos principais objetivos deste trabalho. Além disso, como o consumo de energia elétrica é responsável pela emissão de gases de efeito estufa, dentre os quais o dióxido de carbono (CO_2) é o mais expressivo, outro objetivo deste trabalho é a investigação da emissão de CO_2 atrelada a execução desses algoritmos. Os resultados são comparados com [Strubell et al. 2019], que gerou certa controvérsia na comunidade científica por afirmar que treinar um único modelo de IA pode emitir tanto carbono quanto cinco carros durante suas vidas útil. Como será detalhado na Seção 2, poucos trabalhos analisam o consumo de energia dos algoritmos de AM e a maioria avaliam um conjunto específico de algoritmos, principalmente os de redes neurais profundas. Sendo assim, o principal diferencial deste trabalho está em avaliar algoritmos de diferentes abordagens de AM, além de algoritmos de RNAs. Além disso, com exceção de [Strubell et al. 2019] não foram encontrados trabalhos que avaliam a emissão de CO_2.

O trabalho está organizado em: Seção 2 com alguns trabalhos relacionados; Seção 3 com a metodologia utilizada para a execução dos experimentos; Seção 4 com os experimentos realizados e os resultados obtidos e Seção 5 com considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

O principal foco dos trabalhos apresentados nesta seção são aqueles que avaliam o impacto da execução de algoritmos de IA no consumo energético e, consequentemente, na emissão de CO_2 equivalente (CO_2e). Porém, a literatura nesta área ainda é escassa. O que se encontra são trabalhos que avaliam o desempenho (acurácia e tempo de execução, por exemplo) de diferentes algoritmos de AM para resolver tarefas específicas em uma área de aplicação [Malakar et al. 2018, Olson et al. 2017, Serpa et al. 2018] ou trabalhos que utilizam os algoritmos de AM para predizer o desempenho e o consumo de energia para execução de uma aplicação [Ferreira et al. 2017, Wu et al. 2016, Klôh et al. 2019].

Poucos trabalhos avaliam o consumo de energia dos algoritmos de AM [Li et al. 2016, Yang et al. 2017, García-Martín et al. 2019]. [García-Martín et al. 2019] apresenta diferentes abordagens para estimar consumo energético dos algoritmos *Hoeffding Tree* e *Very Fast Decision Tree* quando executados nas arquiteturas ARM, GPU e X86. Os autores fazem um levantamento de ferramentas que fornecem valores de estimativa de energia, identificando quais componentes podem ser monitorados e em que fase (treinamento ou teste) ocorreu o monitoramento. Porém, ainda que o trabalho seja relevante, são avaliados apenas algoritmos de AD. Atualmente, existem outros algoritmos de AM que são amplamente utilizados para a solução de diversos problemas e, por isso, também necessitam de análise.

No trabalho de [Li et al. 2016] é realizado um estudo sobre a eficiência energética dos modelos de Redes Neurais Convolucionais (RNC). Os autores analisam qual parte do algoritmo consome mais energia e qual a influência das configurações de hardware (CPU e GPU) no consumo energético. São limitados o uso de núcleos dos processadores, a frequência de memória e outros parâmetros referentes à placa de vídeo. Além disso, como existe um grande número de *frameworks* (Caffe, Tensorflow e Torch) para execução de algoritmos de AM, também avaliam o desempenho desses *frameworks* de acordo com as configurações de hardware.

Em [Strubell et al. 2019], um dos motivadores desta pesquisa, foi realizada uma

avaliação do ciclo de vida de modelos de IA, no qual os autores mencionam que o processo de treinamento pode emitir mais de 284000 kg de CO_2e (CO_2 equivalente, pois computadores não emitem CO_2). Ou seja, treinar um único modelo de IA pode emitir tanto carbono quanto cinco carros durante suas vidas útil. Ainda, os custos computacionais e ambientais do treinamento crescem proporcionais ao tamanho do modelo, e este custo é ainda maior quando ajuste de parâmetros nos algoritmos são realizados para aumentar a precisão final dos modelos. Porém, esses resultados foram obtidos para alguns modelos específicos de Processamento de Linguagem Natural (NLP), subárea da IA que teve avanços significativos nos últimos tempos graças a alta capacidade computacional disponível e a novos modelos de redes neurais profundas.

Portanto, vale ressaltar que a maioria dos trabalhos avaliam um conjunto específico de algoritmos de AM, principalmente os de redes neurais profundas. Como mencionado, existem outros algoritmos de AM (tais como Máquinas de Vetores de Suporte, AD, Kmeans, entre outros) que são amplamente utilizados para a solução de diversos problemas, o que os torna relevantes para análise. Com exceção do trabalho de [Strubell et al. 2019], não foram encontrados trabalhos que avaliam a emissão de CO_2e para a execução dos algoritmos de AM.

3. Metodologia de Experimentos

Nesta seção é apresentada a metodologia adotada para a avaliação do consumo de energia em diferentes arquiteturas computacionais durante o treinamento de diversos algoritmos de AM. Além disso, o objetivo também é de avaliar o impacto da emissão de CO_2e desses ambientes, conforme as motivações apresentadas na Seção 1. É importante ressaltar que não houve ajuste dos parâmetros de treinamento pois este trabalho não tem como finalidade obter os melhores modelos em cada tarefa.

São avaliadas diferentes abordagens (supervisionada e não supervisionada) e tarefas de AM (classificação, clusterização e regressão, conforme hierarquia apresentada na Figura 1. Para a tarefa de classificação foi executado um algoritmo de AD na versão C4.5 [Quinlan 1996] e dois para Redes Neurais Artificiais (RNAs): *Multilayer Perceptron* (MLP) e Redes Neurais Convolucionais (RNC). Para regressão é utilizado o algoritmo Floresta Randômica Regressiva (FRR) e para clusterização o KMeans. O objetivo é avaliar algoritmos das diferentes abordagens de AM existentes. Como mencionado, a maioria dos trabalhos se concentram apenas em algoritmos de redes neurais profundas, porém, existem outros algoritmos de AM que também são amplamente utilizados.

Os algoritmos, exceto AD[3], e os conjuntos de dados de treinamento foram obtidos do repositório *Super Data Science*[4] (SDS), com exceção dos conjuntos de dados para as RNAs, os quais foram obtidos do repositório UCI [Asuncion and Newman 2007][5], por apresentarem um tamanho maior para a análise. Não houve alteração nos parâmetros padrão dos algoritmos. O conjunto de dados utilizado pelos algoritmos AD, FRR e Kmeans é composto por, aproximadamente, 4900 exemplos e 12 atributos. Este conjunto de dados é para classificação da qualidade de vinhos. Para o algoritmo MLP, o conjunto de dados possui 10000 exemplos e 14 atributos, os quais classificam clientes de um banco. Para a RNC há 10000 exemplos para classificação de imagens de cães e gatos.

As arquiteturas utilizadas neste trabalho estão detalhadas na Tabela 1. ARQ1 refere-se a uma arquitetura ARM de baixo consumo, ARQ2 é uma arquitetura x86 de alto desempenho

[3]Disponível em: https://github.com/serengil/chefboost

[4]https://www.superdatascience.com/pages/machine-learning

[5]https://archive.ics.uci.edu/ml/datasets/Wine+Quality

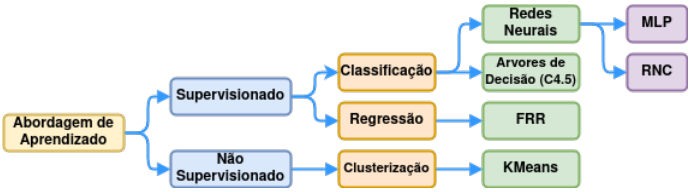

Figura 1. Hierarquia de abordagens de Aprendizado de Máquina

com um processador Intel de Oitava Geração, enquanto ARQ3 possui as mesmas características da ARQ2, mas com uma GPU Nvidia GeForce da família Turing.

	ARQ1	**ARQ2**	**ARQ3**
CPU	NVIDIA Denver 2, Cortex-A57 @ 2GHz	Intel Core i7 8700 @ 3.2GHz	Intel Core i7 8700 @ 3.2GHz
CPU Cores	2C, 4C	6C 12T	6C 12T
RAM	8GB	64GB	64GB
GPU	NVIDIA Pascal @ 1300MHz	-	Nvidia GeForce RTX 2080Ti @1545 MHz
GPU RAM	-	-	11GB
GPU Cores	256C	-	544C
OS	Ubuntu 18.04		
Kernel	4.9	5.3	5.3

Tabela 1. Arquiteturas utilizadas para a execução dos experimentos

Para o cálculo da emissão de CO_2e foram utilizadas as Equações 1 e 2, baseadas no trabalho de [Strubell et al. 2019]. Na Equação 1 é calculada a Energia Total (ET) em KWh, onde PUE representa a eficiência energética, t, o tempo (horas), P_CPU , P_GPU e P_ MEM, respectivamente, potência (Watts) da CPU, GPU e memória RAM. As métricas foram coletadas com a ferramenta perf [6]. Na ARQ2 e ARQ3 não foi possível obter o valor da variável P_ MEM dada a limitação da ferramenta de coleta utilizada. Para estes casos a ET foi considerada apenas sobre o consumo dos núcleos de CPU e GPU, desconsiderando o fator de memória. Na Equação 2 é calculada a emissão de CO_2e, os parâmetros usados são: a constante de CO_2e do país e a ET. O valor da constante para o Brasil é de 0,125 Kg/KWh [Miranda 2012].

$$ET = (PUE \times t \times (P\_CPU + P\_GPU + P\_MEM))/1000 \quad (1)$$

$$CO2e = 0,125 \times ET \quad (2)$$

A eficiência energética de cada algoritmo foi avaliada pela relação consumo energético e tempo de execução usando a métrica *Energy Delay Product* ($EDP = Energia$ (Joules)$\times Delay$ (segundo)), onde *Delay* é o tempo de execução do algoritmo. O tempo de execução e o consumo de energia são coletados para todos os algoritmos em todas as arquiteturas, quando possível. Os algoritmos AD, KMeans e FRR não foram executados na ARQ3, pois as versões para GPU não estavam disponíveis. Todos os experimentos foram executados 30 vezes para garantir uma melhor avaliação dos resultados, os quais são apresentados na Tabela 2.

4. Experimentos e Resultados

Nesta seção são apresentados os experimentos e resultados obtidos na avaliação do consumo de energia e emissão de CO_2 das diferentes arquiteturas computacionais.

[6]Os detalhes de utilização da ferramenta estão disponíveis em [Klôh et al. 2019]

Algoritmos	Potência (Watts)			Tempo (Segundos)		
	ARQ1	ARQ2	ARQ3	ARQ1	ARQ2	ARQ3
AD	4,013	23,584	-	94,724	0,392	-
FRR	4,016	33,194	-	4,537	0,671	-
Kmeans	4,079	56,182	-	11,193	1,988	-
MLP	4,504	40,086	80,790	1624,18	47,914	126,579
RNC	4,098	28,083	78,711	9893,9	1645,275	1452,848

Tabela 2. Média da potência e do tempo de execução dos algoritmos.

4.1. Avaliação do Consumo de Energia

Os resultados, em escala logarítmica, da eficiência energética dos algoritmos KMeans, AD e FRR nas arquiteturas ARQ1 e ARQ2 estão na Figura 2. É possível observar que nos três algoritmos o menor EDP foi obtido na ARQ2, demonstrando assim uma eficiência energética melhor. Apesar dos algoritmos FRR e KMeans terem consumido mais energia na ARQ2 do que na ARQ1 (uma arquitetura de baixo consumo do tipo ARM), o tempo de execução muito maior na ARQ1 influenciou no EDP (Tabela 2). Além disso, para o algoritmo AD, tanto o tempo de execução quanto o consumo de energia foram muito maiores na ARQ1, apresentando assim o menor *EDP* quando executado na arquitetura ARQ2 (Figura 2).

Para os experimentos com os algoritmos MLP e RNC, além da execução nas três arquiteturas, também foram utilizados dois tipos de Tensorflow: o da Google [Abadi et al. 2016] (TFG) e o da Intel [Ould-Ahmed-Vall 2017] (TFI). O TFG é um framework padrão desenvolvido com o intuito de simplificar o desenvolvimento de redes neurais utilizando a linguagem de programação Python. O Tensorflow da Intel, desenvolvido em parceria com a Google, permite que os processadores Xeon e Xeon Phi possam aproveitar o máximo de desempenho das instruções de vetor (AVX 2/AVX 512), as quais são amplamente utilizadas por algoritmos de redes neurais. O objetivo é comparar o desempenho dos diferentes tipos de Tensorflow e avaliar se o uso pode reduzir o consumo de energia para o treinamento das RNAs.

Na Figura 3 são apresentados os resultados do EDP para os algoritmos MLP e RNC, quando executados nas arquiteturas ARQ1, ARQ2 e ARQ3, com o TFG. Além disso, para a ARQ2 também foi realizado o experimento utilizando o TFI, identificado no gráfico como *ARQ2. Como pode ser observado, para a rede MLP, o TFG obteve um melhor resultado, enquanto que para a RNC, foi o TFI. Esse resultado é devido a diferença do conjunto de dados utilizado pelas redes MLP (atributos correspondem a valores numéricos) e RNC (imagens). Ou seja, a RNC sendo treinada para classificar imagens se beneficia com técnicas de vetorização, como as que são utilizadas pelo TFI.

4.2. Avaliação da Emissão de CO_2e

Na Figura 4 é apresentado o resultado das emissões de CO_2e (escala original) das técnicas FRR e Kmeans nas diferentes arquiteturas. A arquitetura ARQ1 apresenta uma emissão menor de CO_2e. Isso pode ser explicado pelo menor consumo de potência em relação as demais arquiteturas. Para a AD, mesmo com o consumo de potência baixo, o tempo de execução foi muito alto. É interessante notar que nos algoritmos FRR e Kmeans, as diferenças do EDP entre as ARQ2 e ARQ1 não são tão significativas, mas para a AD, essa diferença é muito discrepante. Para os algoritmos de RNAs as emissões de CO_2e são apresentados na Figura 5 (escala original). Conforme visto nos outros algoritmos, a relação entre CO_2e e potência consumida também acontece com as RNAs. Novamente a ARQ1 teve melhor resultado na emissão de CO_2e em relação as outras arquiteturas. Porém, este resultado não foi obtido para o algoritmo MLP, onde a ARQ2 obteve melhor resultado tanto de EDP quanto de emissão de CO_2e. Isso ocorreu porque os novos processadores da Intel vem com instruções especificas para cálculos vetoriais.

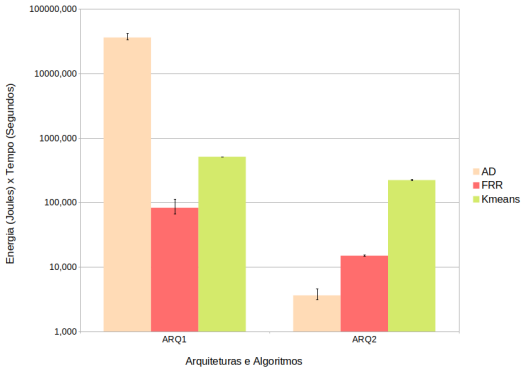

Figura 2. EDP dos algoritmos AD, FFR e Kmeans.

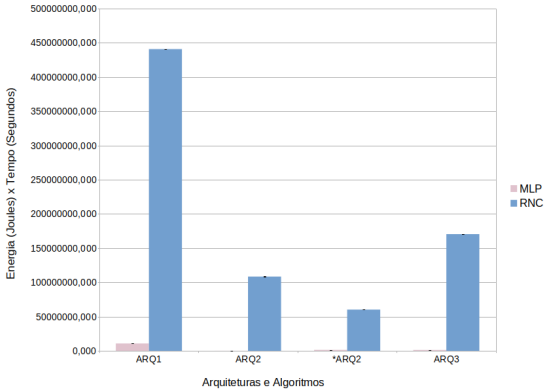

Figura 3. EDP dos algoritmos MLP e RNC.

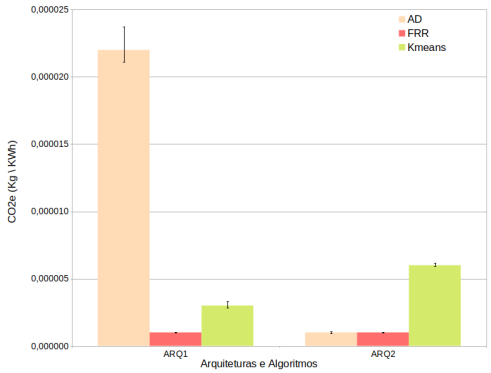

Figura 4. CO_2e dos algoritmos AD, FFR e Kmeans.

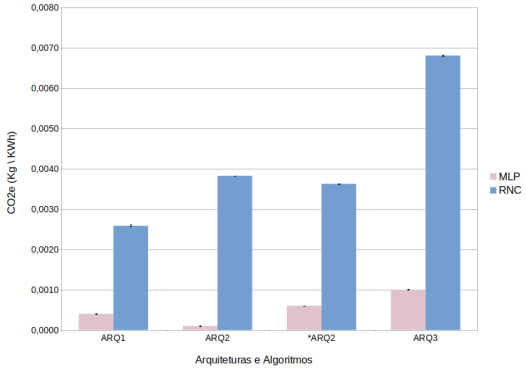

Figura 5. CO_2e dos algoritmos MLP e RNC.

Para efeito comparativo, o carro popular mais vendido no Brasil, segundo o Inmetro, emite 0,100 Kg/km de CO_2 (motor 1.0 - 8 V, gasolina). Para que a arquitetura utilizada neste trabalho possa emitir 0,100 Kg de CO_2e são necessárias 5,74 horas de execução contínua do algoritmo RNC, na ARQ3. Apesar dos valores encontrados neste experimento não serem tão expressivos quanto os valores de [Strubell et al. 2019], é importante observar que o algoritmo de RNC foi apenas treinado, sem ajuste de parâmetros para se chegar a um modelo com precisão aceitável. Normalmente são necessários vários ajustes de parâmetros e muitas execuções de treinamento. Além disso, o conjunto de treinamento utilizado neste trabalho para o algoritmo RNC é pequeno (10000 exemplos), por isso o tempo de treinamento é baixo (1452s), o que para um problema real (com conjuntos de dados bem maiores) pode facilmente chegar a mais de 100h. Outro fator que impacta neste resultado é a constante de CO2e utilizada no cálculo, a qual muda de acordo com o país onde é executado o treinamento.

Usinas de energia a carvão podem emitir uma quantidade muito mais expressiva de CO_2e do que as hidrelétricas. É possível comparar a emissão de CO2e entre o Brasil, China e EUA para executar uma RNC na ARQ3 (Tabela 3) alterando o valor da constante de CO2e do país, utilizada na Equação 2. Para os três países, a maior emissão de CO2e seria da China, o que está diretamente relacionado ao carvão e assim a constante de CO2e.

5. Considerações Finais e Trabalhos futuros

Este trabalho apresenta a avaliação do desempenho computacional e do consumo energético feito por meio do EDP e da emissão de CO_2e para diferentes arquiteturas computacionais, utili-

País	CO_2 equivalente (Kg)
Brasil	0,009
China	0,037
EUA	0,026

Tabela 3. CO_2e emitido pela RNC quando executada na ARQ3 no Brasil, e caso fosse executada na China e EUA com outras matrizes energéticas.

zando diferentes algoritmos de AM. Um dos objetivos é a investigação dos resultados apresentados no trabalho de [Strubell et al. 2019], que demonstra que o desenvolvimento de um modelo final de IA (para NLP) pode ser tão poluente quanto cinco carros. Apesar dos resultados encontrados neste trabalho não serem tão altos, ainda assim são relevantes, pois facilmente uma RNC é executada por mais de 5h em experimentos reais, o que equivale ao consumo de um carro para trafegar 1km. Além disso, esses resultados são da execução em uma única GPU para a matriz energética brasileira, predominantemente hidrelétrica, e por isso menos poluente. É importante observar que cada arquitetura possui suas próprias vantagens que variam entre consumo energético, tempo de execução e emissão de CO_2e, cabendo ao usuário priorizar qual arquitetura atende suas necessidades. Algoritmos executados em CPU tem um melhor equilíbrio entre EDP e emissão de CO_2e em relação a outras arquiteturas, mantendo os dois em níveis relativamente baixos, especialmente para AD, FRR e em ambos os algoritmos de RNAs. Quanto à emissão de CO_2e, arquiteturas baseadas em ARM tem o nível mais baixo de emissão para os algoritmos KMeans, FRR e RNC, mas tem o pior EDP em todos os casos. Isso se deve à natureza da arquitetura, a qual prioriza o equilíbrio energético, mas mantém a capacidade computacional suficiente para todos os tipos de trabalhos. Apesar das GPUs apresentarem menor tempo de execução para as RNAs, dado o número de núcleos especializados para esse tipo de tarefa, essa arquitetura tem o pior EDP entre todas as arquiteturas, sendo ainda mais ineficiente quando se considera o nível de emissão de CO_2e. Isso é consequência do alto consumo energético das GPUs. Além disso, através dos resultados com diferentes tipos de Tensorflow, foi demonstrado a importância da otimização dos algoritmos de AM para as diferentes arquiteturas.

Este trabalho é um estudo inicial sobre o tema e, ainda está em desenvolvimento. Diante das limitações observadas até o momento, como trabalhos futuros são propostas: explorar outros algoritmos de RNAs e outras arquiteturas; a possibilidade de paralelizar os algoritmos para executá-los em clusters; reproduzir testes já publicados na literatura para fins de comparação com outras arquiteturas; realizar experimentos com algoritmos em diferentes linguagens de implementação; e comparar qual o impacto na utilização de APIs, Bibliotecas e Frameworks (tais como Pytorch, Keras, MXNET, Spark MLlib, entre outros) na emissão de CO2 e consumo de energia.

Agradecimentos

Os autores agradecem ao CNPq e a FAPERJ pelo apoio financeiro.

Referências

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Asuncion, A. and Newman, D. (2007). Uci machine learning repository. https://archive.ics.uci.edu/ml/index.php.

Avelar, V., Azevedo, D., French, A., and Power, E. N. (2012). Pue: a comprehensive examination of the metric.

Ferreira, A. R. et al. (2017). Um modelo analítico para estimar o consumo de energia de sistemas multi-camadas no nível de transação. Master's thesis, Universidade Federal de Goiás.

García-Martín, E., Rodrigues, C. F., Riley, G., and Grahn, H. (2019). Estimation of energy consumption in machine learning. *Parallel and Distributed Computing*, 134:75–88.

Google (2019). Google 2019 environmental web report. https://sustainability.google/reports/environmental-report-2019.

Klôh, V., Gritz, M., Schulze, B., and Ferro, M. (2019). Towards an autonomous framework for hpc optimization: Using machine learning for energy and performance modeling. In *Anais do XX Simpósio em Sistemas Computacionais de Alto Desempenho*, pages 438–445, Porto Alegre, RS, Brasil. SBC.

Li, D., Chen, X., Becchi, M., and Zong, Z. (2016). Evaluating the energy efficiency of deep convolutional neural networks on cpus and gpus. In *2016 IEEE International Conferences on Big Data and Cloud Computing, Social Computing and Networking, Sustainable Computing and Communications*, pages 477–484. IEEE.

Malakar, P., Balaprakash, P., Vishwanath, V., Morozov, V., and Kumaran, K. (2018). Benchmarking machine learning methods for performance modeling of scientific applications. In *2018 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, pages 33–44.

Miranda, M. M. d. (2012). *Fator de emissão de gases de efeito estufa da geração de energia elétrica no Brasil: implicações da aplicação da Avaliação do Ciclo de Vida*. PhD thesis, Universidade de São Paulo.

Olson, R. S., La Cava, W., Orzechowski, P., Urbanowicz, R. J., and Moore, J. H. (2017). Pmlb: a large benchmark suite for machine learning evaluation and comparison. *BioData mining*, 10(1):1–13.

Ould-Ahmed-Vall, E. (2017). Tensorflow optimizations on modern intel architecture. https://software.intel.com/en-us/articles/tensorflow-optimizations-on-modern-intel-architecture.

Quinlan, J. R. (1996). Improved use of continuous attributes in c4.5. *Journal of artificial intelligence research*, 4:77–90.

Santos, L. d. O. (2015). *Caracterização de tarefas usando Redes Neurais e CUDA*. PhD thesis, Universidade Estadual Paulista "Júlio de Mesquita Filho.

Serpa, M. S., Krause, A. M., Cruz, E. H., Navaux, P. O. A., Pasin, M., and Felber, P. (2018). Optimizing machine learning algorithms on multi-core and many-core architectures using thread and data mapping. In *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 329–333. IEEE.

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650.

Wu, X., Taylor, V., Cook, J., and Mucci, P. J. (2016). Using performance-power modeling to improve energy efficiency of hpc applications. *Computer*, 49(10):20–29.

Yang, T.-J., Chen, Y.-H., and Sze, V. (2017). Designing energy-efficient convolutional neural networks using energy-aware pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5687–5695.