Consumo de energia e emissão de poluentes na implementação de modelos de aprendizado de máquina

Introdução

A recente expansão do mercado de modelos de aprendizado de máquina (ML — machine learning) foi baseada não só no desenvolvimento de modelos mais sofisticados para resolver problemas complexos, como o uso de Transformers para problemas de processamento de linguagem natural e geração de imagens, mas no aumento quantitativo no número de parâmetros dos modelos.
Para servir modelos grandes a cada vez mais usuários, novos datacenters estão sendo construídos no mundo todo, tornando necessário analisar o impacto ecológico dessa expansão.
Nesse trabalho, nos preocupamos em estudar especificamente o gasto de energia e a consequente emissão de gases poluentes pelas implementações de modelos de ML.

Carbono incorporado

Na literatura, é comum encontrar a divisão das formas de emissão entre dois tipos:

- o carbono incorporado (*embodied carbon*), correspondente às emissões que ocorrem durante o processo de produção das peças de *hardware* que compõem um computador ou data center.
- o carbono operacional (*operational carbon*), correspondente às emissões resultantes do funcionamento do *hardware*, seja executando o treinamento e inferência de modelos ou em períodos ociosos.

(Wu et al., 2022) estimaram que o carbono incorporado representava 30% das emissões nas implementações de grandes modelos de ML no Facebook (atual Meta) em 2020. Essa proporção é maior para servidores e computadores que utilizam energia mais limpa (ou menor intensidade de carbono) e celulares.
Calcular o carbono incorporado é um processo complexo, pois depende do fornecimento de dados precisos por parte de todas as partes envolvidas no processo produtivo.
(Gupta et al., 2022) propôs o modelo ACT, que estima:

- as emissões E de CPUs, GPUs e outras unidades de processamento a partir do tipo de pastilha (CPA — carbono por área) e a eficiência (Y — yield) na produção do semicondutor

$$E = \frac{1}{Y} * \text{área} * CPA$$

- as emissões E de HDDs, SSDs e outros componentes de armazenamento pela quantidade de carbono emitido por byte a ser armazenado (CPB)

$$E = \text{capacidade em bytes} * CPB$$

Os valores para CPA e CPB são obtidos, a princípio, a partir de dados publicados pelos fabricantes. Vale ressaltar que o "carbono" aqui representa os vários gases causadores de efeito estufa emitidos ou utilizados durante a produção de peças de *hardware*, como o trifluoreto de nitrogênio (NF_3) e os perfluorocarbonetos (PFCs).
No que tange as unidades de processamento, GPUs têm carbono incorporado maior que CPUs, por terem áreas maiores. Já nos componentes de armazenamento, SSDs têm maior CPB que HDDs. Portanto, é necessário encontrar um balanço na arquitetura de *datacenters* de forma a utilizar cada componente o mais eficientemente possível, minimizando emissões desnecessárias.

Carbono operacional

O carbono operacional tem alta relação com o tempo de execução dos programas, podendo ser estimado a partir do número de operações de ponto flutuante (FLOPs — *floating point operations*) realizadas durante a execução. (Faiz et al., 2024) proporam o modelo LLMCarbon, projetado para modelos de linguagem mas extensível para outros modelos.
Primeiro, calculamos a utilização de FLOPs pelo modelo (MFU — *model FLOP utilization*), métrica de eficiência proposta por (Chowdhery et al., 2023)

$$MFU = \frac{F * T}{M}$$

Onde:

- F — FLOPs por *token* gerado
- T — *tokens* gerados por segundo
- M — máximo de FLOPs por segundo que podem ser executados pelo *hardware*

Depois, as emissões E podem ser estimadas como:

$$E = W * MFU * s * N * PUE * IC$$

Onde:

- W — potência máxima da unidade de processamento (CPU, GPU, FPGA etc.), em watts
- s — tempo de execução em segundos
- N — número de unidades de processamento
- PUE — eficiência no uso de energia (*power usage efficiency*), razão entre o total de energia gasto pelo *datacenter* e o total gasto pelos computadores em si
- IC — intensidade de carbono da fonte de energia, ou seja, a quantidade de gramas de CO_2 emitido por watt-segundo de energia gerado.

Formas de redução do carbono operacional

Baseados em reduzir o número de FLOPs realizados:

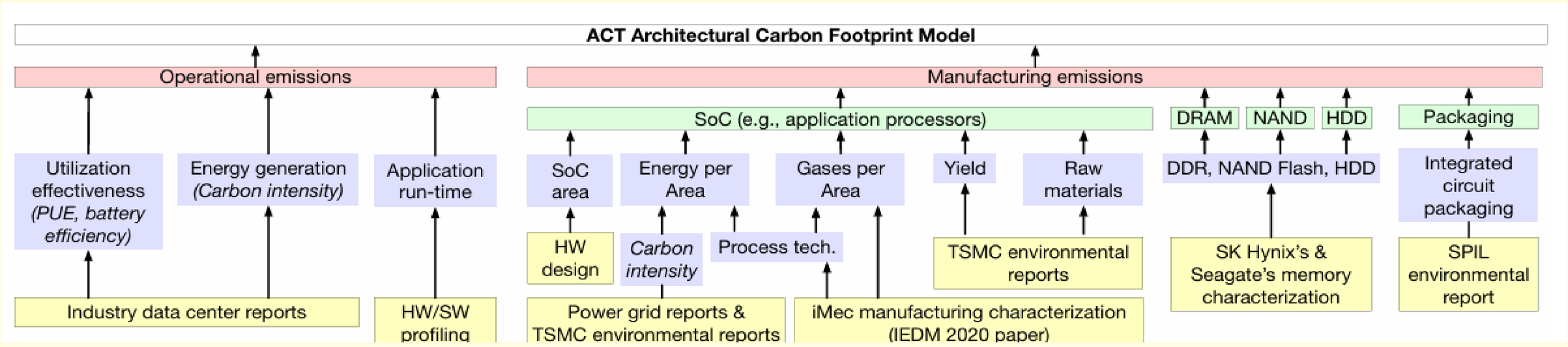
- Escolher hiperparâmetros que realizam menos operações
- Avaliar custo-benefício de entre precisão/acurácia e tamanho dos modelos
- Técnicas de redução do conjunto de dados de treinamento, como pular *mini-batches* de exemplos estocasticamente, como proposto por (Wang et al., 2019)

Baseados em otimizar a utilização do *hardware*:

- Otimizar a distribuição dos parâmetros na memória das GPUs e afins para minimizar o número de carregamentos
- Combinar operações consecutivas em um único comando para a GPU
- Analizar custo-benefício entre a frequência operante das GPUs e a latência
- Particionar GPU entre modelos semelhantes porém tamanhos distintos (modelos de qualidade mista — *mixed quality*, como em (Li et al., 2023)

Também é essencial maximizar a utilização de energia com menor intensidade de carbono, como a energia eólica, nuclear ou solar, além de maximizar o PUE das instalações, com melhores técnicas de resfriamento e de contenção do calor gerado pelo *hardware*

Diagrama do modelo ACT (Gupta et al., 2022)



Conclusão

Aferir o consumo de energia e o impacto ecológico das recentes evoluções no mercado de ML é um processo ainda bastante difícil, pois depende de informações que as maiores empresas do meio têm evitado em divulgar, seja os dados de consumo em si ou dados que permitam estimar as emissões, como o número de parâmetros dos modelos, hardware utilizado e intensidade de carbono das fontes de energia.
Por isso, é necessário que haja maior transparência por parte dessas empresas na divulgação desses dados e de novas descobertas que permitam otimizar a implementação dos modelos, reduzindo tempos de processamento e, consequentemente, o gasto energético.

Bibliografia

- Chowdhery, Aakanksha et al. (Jan. 2023). "PaLM: scaling language modeling with pathways". *J. Mach. Learn. Res.* 24.1. ISSN: 1532-4435.
- Faiz, Ahmad et al. (2024). *LLMCarbon: Modeling the end-to-end Carbon Footprint of Large Language Models*. arXiv: 2309.14393 [cs.CL]. URL: https://arxiv.org/abs/2309.14393.
- Gupta, Udit et al. (2022). "ACT: Designing Sustainable Com puter Systems With An Architectural Carbon Modeling Tool". In: *Proceedings of The 49th Annual International Symposium on Computer Architecture (ISCA '22)*. ACM.
- Li, Baolin et al. (Nov. 2023). "Clover: Toward Sustainable AI with Carbon-Aware Machine Learning Inference Service". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC '23. ACM, pp. 1–15. DOI: 10.1145/3581784.3607034. URL: http://dx.doi.org/10.1145/3581784.3607034.
- Wang, Yue et al. (2019). *E2-Train: Training State-of-the-art CNNs with Over 80% Energy Savings*. arXiv: 1910.13349 [cs.LG]. URL: https://arxiv.org/abs/1910.13349.
- Wu, Carole-Jean et al. (2022). *Sustainable AI: Environmental Implications, Challenges and Opportunities*. arXiv: 2111.00364 [cs.LG]. URL: https://arxiv.org/abs/2111.00364.

Luca Diogo da Silva
Orientador: Prof. Dr. Alfredo Goldman vel Lejbman
Trabalho de Conclusão de Curso - MAC0499