Introduction to Computational Sustainability

Lecture 2: Large-Scale AI Models and Sustainability Aspects + Assignment 1 Specs

Paul Joe Maliakel

Credits for most of the material used in this presentation go to Dr. Shashikant Ilager & Dr. Alessandro Tundo.

What Is A Large-Scale AI Model?

AI models are built upon a wide range of algorithms and techniques, e.g., regression models, ensemble models, (deep) neural networks.

A **large-scale AI model** is a model that:

- it is trained on very large datasets
- it has a huge size
- it is composed by complex architectures with billions of parameters
- it is trained high-powered computational resources

However, no single & accepted definition of when it can be classified as "large-scale"

Here, we focus on their impact on sustainability, in particular:

1. **energy consumption** and **carbon footprint** during both training and inference
2. **mitigation strategies**

Examples of Large-Scale Models

- Multi-modal (e.g., Dall-E, Chamaleon, Qwen2-VL)
- Large Language Models (LLMs) (e.g., GPT, PaLM, Llama, Grok)
- ...the next one they will release tomorrow :-)

"a teddy bear on a skateboard in times square"

"a painting of a fox sitting in a field at sunrise in the style of Claude Monet"

Source: OpenAI.com

The recent breakthroughs in AI are achieved by sheer scale rather than new algorithmic techniques!

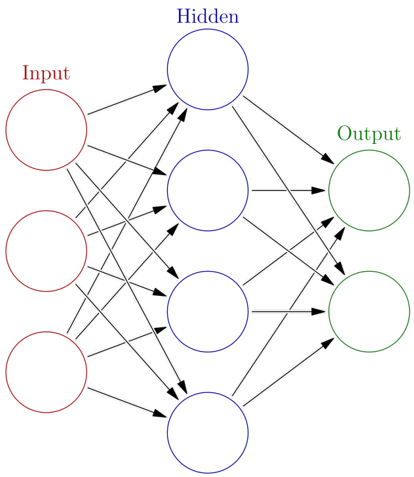
Why Is It More the Scale?

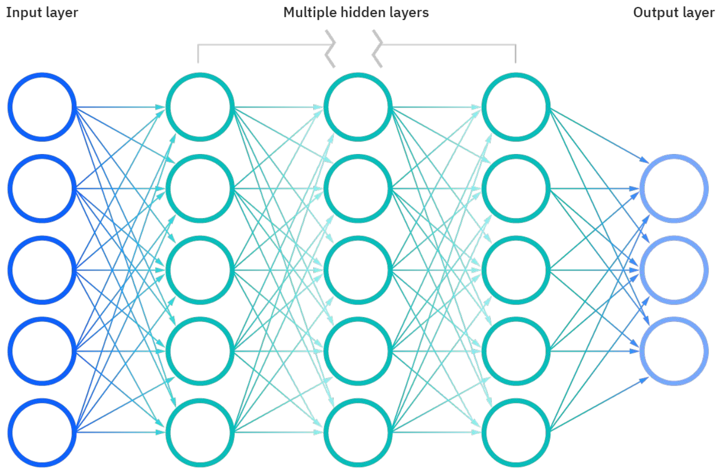
- Most of the algorithms are there since a while...
  - Transformer-based models (e.g., GPT) is deep learning technique published in 2017
  - Recurrent Neural Networks (RNNs) have been revamped in the 80-90s, but they are even older!
  - Neural Networks (NNs) have been in general started be the practical applied in the 2000s

*It is in the possibility to train on a **massive amount of data**, using data centers composed of GPUs and other **accelerators**, disposable **cloud resources**, and **industry attention** that made possible the current "AI boom"!*

From Artificial (Shallow) NNs to Deep NNs

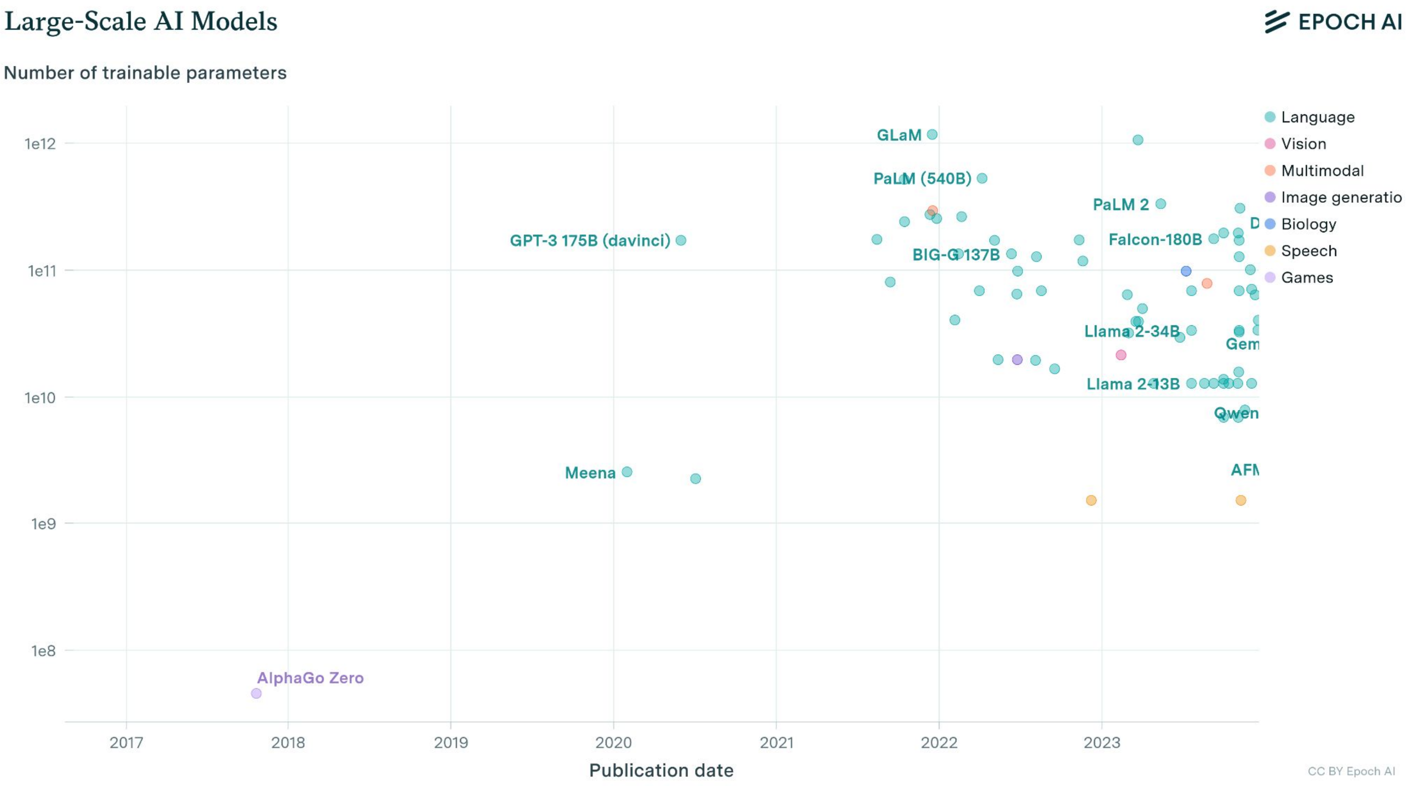
Most of these large-scale models are fundamentally based on Deep Neural Networks (DNNs)





Sources:
- https://commons.wikimedia.org/wiki/File:Colored_neural_network.svg
- https://www.analyticsvidhya.com/blog/2022/11/analyzing-and-comparing-deep-learning-models/

How Large Are the Models?

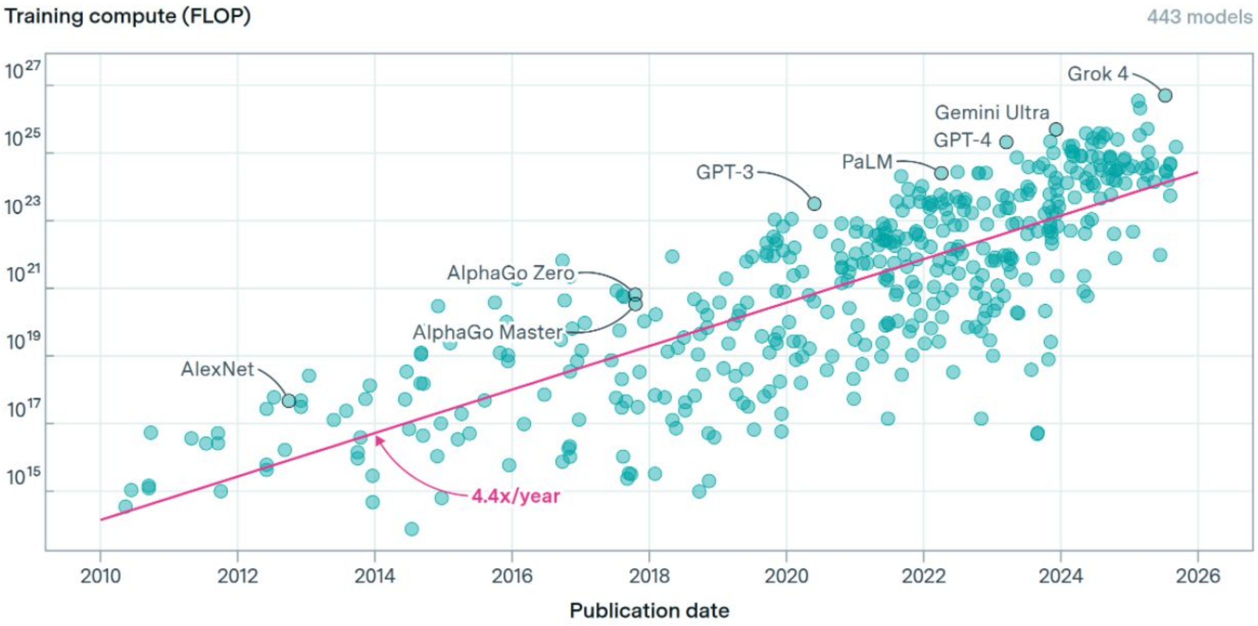


How Large Are the (Language) Models?

System	Domain	Publication date	Parameters	Training compute (FLOP)	Training dataset size (...	Training time (hours)	Training hardware	Hardware quantity	Training compute cost ...
GLaM	Language	2021-12-13	1200000000000.00	3.74e+23	600000000000	1366.0	Google TPU v4	1024	$541,437.42
PanGu-Σ	Language	2023-03-20	1085000000000.00	4.669999999999999e+23	246750000000	2400.0	Huawei Ascend 910	512	
PaLM (540B)	Language	2022-04-04	540350000000.00	2.5272e+24	585000000000	1536.0	Google TPU v4	6144	$2,945,949.76
Megatron-Turing NLG 530B	Language	2021-10-11	530000000000.00	1.17e+24	270000000000	770.0	NVIDIA A100 SXM4 80 GB	4480	$3,704,291.31
MegaScale (530B)	Language	2024-02-23	530000000000.00	9.6910000000001e+23	300000000000	117.9	NVIDIA A100	11200	
MegaScale (Production)	Language	2024-02-23	530000000000.00	1.2e+25		504.0	NVIDIA A100	12288	
Llama 3.1-405B	Language	2024-07-23	405000000000.00	3.8e+25	15600000000000	2142.0	NVIDIA H100 SXM5 80GB	16000	
PaLM 2	Language	2023-05-10	340000000000.00	7.34e+24	2700000000000		Google TPU v4		$4,865,570.06
Nemotron-4 340B	Language	2024-06-14	340000000000.00	1.8000000000000003e+25	6750000000000	2200.0	NVIDIA H100 SXM5 80GB		
Grok-1	Language	2023-11-04	314000000000.00	2.90000000001e+24	6200000000000				
Gopher (280B)	Language	2021-12-08	280000000000.00	6.31e+23	300000000000	920.0	Google TPU v3	4096	$616,611.14
ST-MoE	Language	2022-02-17	269000000000.00	2.9000000000000005e+23	1500000000000				
ERNIE 3.0 Titan	Language	2021-12-23	260000000000.00	1.0421e+24	668000000000		Huawei Ascend 910 NVIDI/	1920	
Yuan 1.0	Language	2021-10-12	245730000000.00	3.5380000000001e+23	1000000000000			2128	
DeepSeek-Coder-V2 236B	Language	2024-06-17	236000000000.00	1.2852e+24	3191000000000				
DeepSeek-V2	Language	2024-05-07	236000000000.00	1.02e+24	8100000000000		NVIDIA H800		
DeepSeek-V2.5	Language	2024-09-06	236000000000.00	1.7892000000000004e+24					
HyperCLOVA 204B	Language	2021-09-10	204000000000.00				NVIDIA A100		
Mi:dm 200B	Language	2023-10-31	200000000000.00	1.2e+24	1000000000000				
Falcon-180B	Language	2023-09-06	180000000000.00	3.76e+24	2625000000000	4320.0	NVIDIA A100 SXM4 40 GB	4096	$10,340,911.71

Source: https://epochai.org/data/large-scale-ai-models?view=table

Computational requirements

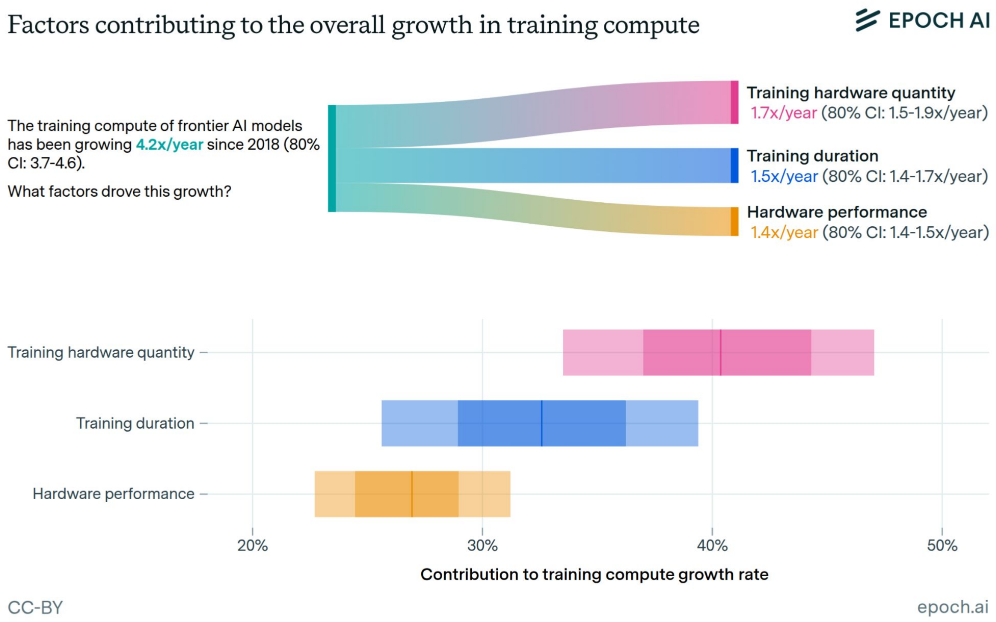


- Since 2010, the training compute used to create AI models has been growing at a rate of 4.4x per year.
- Most of this growth comes from increased spending, although improvements in hardware have also played a role.

The training compute of AI models has been doubling roughly every six months.

Source: https://epochai.org/data/large-scale-ai-models?view=table

Factors contributing

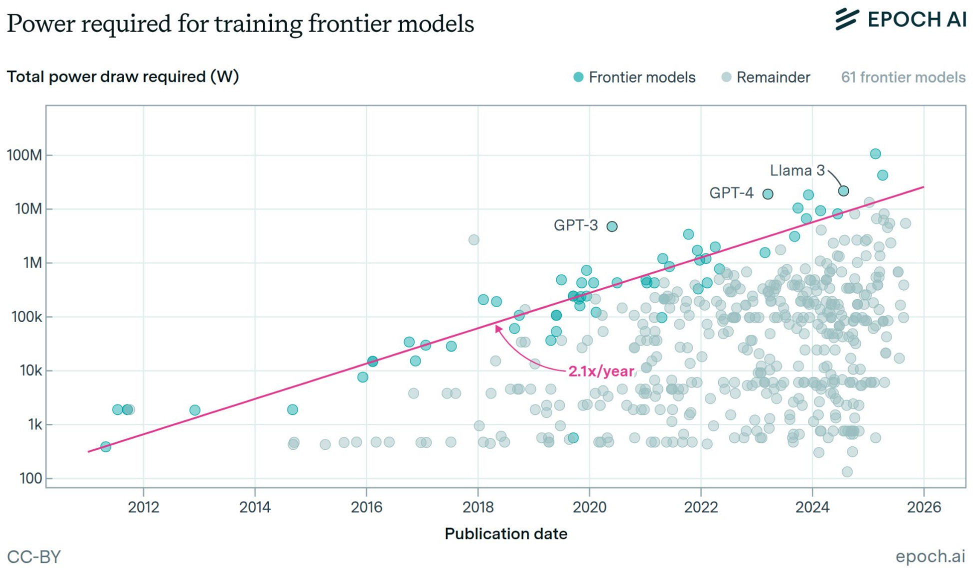


- Since 2018, significant model compute has scaled most likely by using much larger training clusters—plus longer runs and faster chips—fueled by surging investment.
- AI budgets have grown 2–3× yearly, financing vast training/inference clusters and ever-bigger models.

Training compute growth is driven by larger clusters, longer training and better hardware

Source: https://epochai.org/data/large-scale-ai-models?view=table

Power



- power draw per GPU is also growing, but at only a few percent per year.
- However, hardware efficiency (a 12x improvement in the last ten years), the adoption of lower precision formats (an 8x improvement) and longer training runs (a 4x increase) account for a roughly 2x/year decrease in power requirements relative to training compute.

Power required to train AI models is doubling annually.

Source: https://epochai.org/data/large-scale-ai-models?view=table

Environmental Cost of Large-Scale Models

- We have to collect, store, and use a large amount of data…all come with a cost!
- We need powerful data centers and supercomputers
- Using energy creates CO_2, the primary greenhouse gas emitted by humans
- The majority of cloud computing providers' energy is not sourced from renewable sources or still uses brown energy to introduce reliability
- Renewable energy sources are still costly to the environment

Some Facts & Numbers

- Microsoft pumped 11.5 million gallons of water to its cluster of Iowa data centers (DCs), before the OpenAI chatgpt release, creating a water shortage in the district

- Total 1.7 billion gallons of water, across all Microsoft DCs in 2022

- ChatGPT gulps up 500 milliliters of water (close to what's in a 16-ounce water bottle) every time you ask questions with 5 to 50 prompts

- Data centers alone are estimated to consume 1/5th of global electricity generated (more than the airline industry!)

Most people are not aware of the resource usage underlying models like GPT, if we are not aware of the resource usage, then there's no way that we can help conserve the resources!

Sources:

- https://fortune.com/2023/09/09/ai-chatgpt-usage-fuels-spike-in-microsoft-water-consumption/
- Li, P., Yang, J., Islam, M. A., & Ren, S. (2023). Making ai less" thirsty": Uncovering and addressing the secret water footprint of ai models. *arXiv preprint arXiv:2304.03271*.
- https://www.weforum.org/agenda/2024/02/harnessing-waste-energy-data-centres/
- https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks

How Do We Move Towards Greener AI?

We need 3 key ingredients:

1. **Measurements:** e.g., standard metrics
2. **Methods:** e.g., algorithm optimization, efficient hardware
3. **Tools:** e.g., off-the-shelf tools to apply methods

Metrics for Measuring ML Impact

CO_2-equivalent emissions (CO_2e): CO_2 and all the other greenhouse gasses (e.g., methane, nitrous oxide, and so on)

Measure of CO_2: metric tons (tCO_2e), representing 1,000 kg (2,205 lb)

Measure of energy: 1 MWh, representing 1 million W of electricity used continuously for 1 h

Data center efficiency: Power usage effectiveness (PUE), the ratio between total energy use (including all overheads, such as cooling) divided by the computing equipment's energy

Carbon intensity: tCO2e/MWh (metric tons per megawatt hour), measures of the cleanliness of a data center's energy

Source: Patterson, D., Gonzalez, J., Hölzle, U., Le, Q., Liang, C., Munguia, L. M., ... & Dean, J. (2022). The carbon footprint of machine learning training will plateau, then shrink. Computer, 55(7), 18-28.

Example for Model Training

MWh = hours to train × number of processors × average power per processor

…Let’s refine it including data center efficiency (PUE)

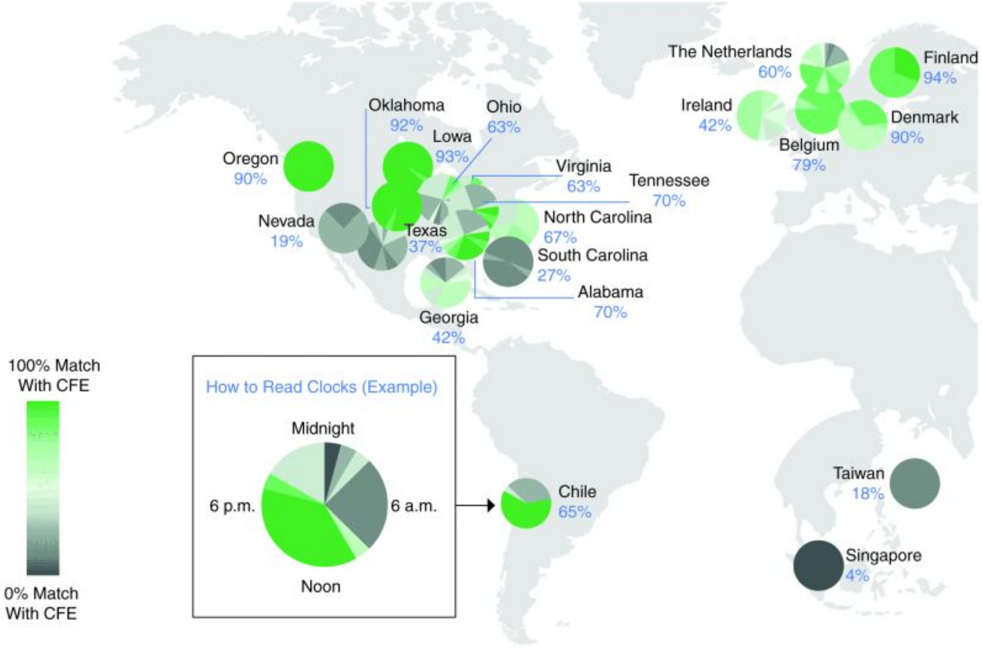
MWh = (hours to train × number of processors × average power per processor) × PUE

…We turn energy into carbon by multiplying it with the carbon intensity of the energy supply

tCO_2e = MWh x tCO_2e per MWh.

Source: Patterson, D., Gonzalez, J., Hölzle, U., Le, Q., Liang, C., Munguia, L. M., … & Dean, J. (2022). The carbon footprint of machine learning training will plateau, then shrink. Computer, 55(7), 18-28.

How Green is The DCs Energy?



- The percentage of CFE by Google Cloud location in 2020
- The map shows the percentage and how it changes by time of day
- Chile has a high CFE percentage from 6 a.m. to 8 p.m. but not at night
- The U.S. examples range from 19% CFE in Nevada to 93% in Iowa, which has strong prevailing winds during the night and day

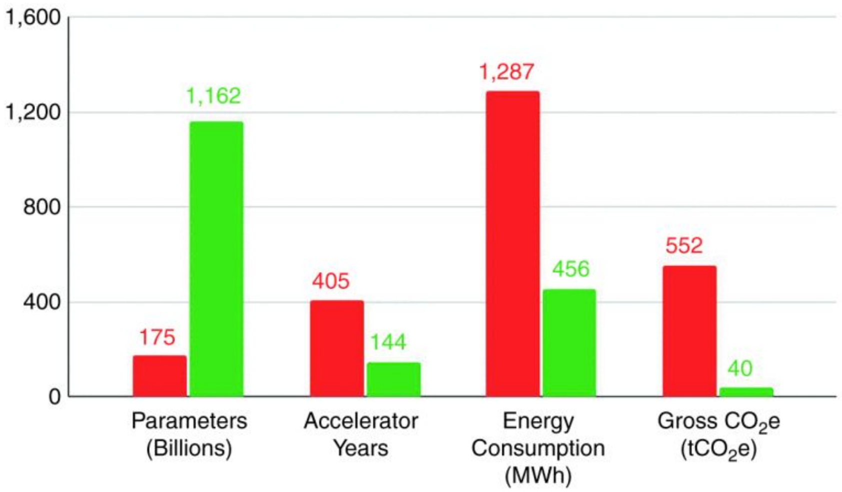
Source: Patterson, D., Gonzalez, J., Hölzle, U., Le, Q., Liang, C., Munguia, L. M., ... & Dean, J. (2022). The carbon footprint of machine learning training will plateau, then shrink. Computer, 55(7), 18-28.

The 4M Practices

1. **Model**: Selecting efficient ML model architectures, such as sparse models versus dense models
   a. can reduce computation by factors of about 5–10x
2. **Machine**: Using processors optimized for ML training, such as tensor processing units (TPUs) and recent GPUs VS general-purpose processors
   a. can improve performance/watt by factors of 2–5.
3. **Mechanization**: Computing in the cloud rather than on-premise
   a. reducing energy costs by a factor of 1.4–2, the cloud DCs are usually more optimized than your premises
4. **Map**: Select the location with the cleanest energy
   a. reducing the gross carbon footprint by factors of 5–10.

Source: Patterson, D., Gonzalez, J., Hölzle, U., Le, Q., Liang, C., Munguia, L. M., ... & Dean, J. (2022). The carbon footprint of machine learning training will plateau, then shrink. Computer, 55(7), 18-28.

Case Study 1: GPT-3 vs GLaM



The parameters, accelerator years of computation, energy consumption, and gross CO2e for GPT-3 (V100 in 2020, in red) and GLaM (TPU v4 in 2021, in green).

Source: Patterson, D., Gonzalez, J., Hölzle, U., Le, Q., Liang, C., Munguia, L. M., ... & Dean, J. (2022). The carbon footprint of machine learning training will plateau, then shrink. Computer, 55(7), 18-28.

Some Modeling Methods...

- Model pruning
- Model **quantization**: reduces memory space by sacrificing some accuracy
- Learning approach: zero-shot vs few-shot learning
- Model cascading

What About Inference?

- Although doesn't consume much energy compared to training, *the sheer number of invocation creates energy bottlenecks*

- Smaller energy footprint, *but billions of invocation*

- Different inference location:
  - Data centers:
    - Enterprise applications
    - Ads serving platforms
    - recommender systems
  - At the Edge:
    - Time-critical applications
    - Smart assistants
    - Connected vehicles
    - Object detection

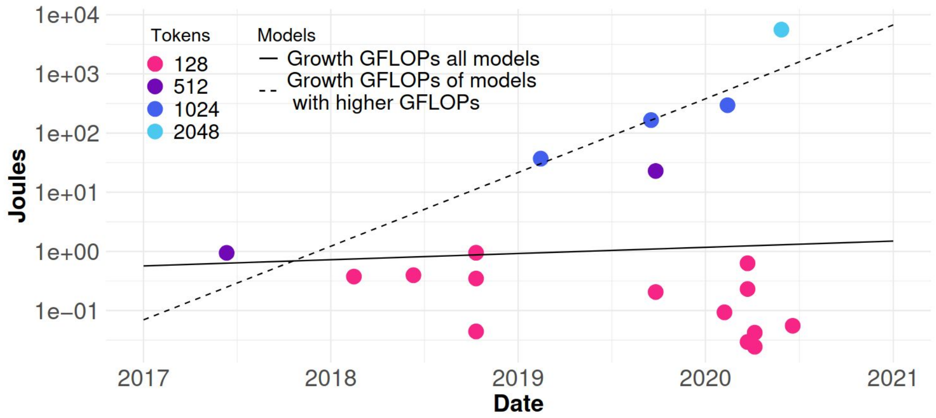
Inference at Edge

- Main challenges:
  - Heterogeneous deployment devices and architectures
  - Limited resources
  - Battery-based or limited power supply
  - Unreliable environments (e.g., network links)
- Optimization methods:
  - Pruning
  - Quantization
  - Cascading
  - context-aware model selection

Estimating Inference Energy Consumption

$$\text{Efficiency} = \frac{\text{HW Perf.}}{\text{Power}} \text{ in units: } \frac{FLOPS}{Watt} = \frac{FLOPs/s}{Joules/s} = \frac{FLOPs}{Joule}$$

$$\text{Energy} = \frac{\text{Fwd. Pass}}{\text{Efficiency}} \text{ in units: } \frac{FLOPs}{FLOPs/Joule} = Joule$$



Estimated Joules of a forward pass (NLP)

Source: Desislavov, R., Martínez-Plumed, F., & Hernández-Orallo, J. (2021). Compute and energy consumption trends in deep learning inference. arXiv preprint arXiv:2109.05472.

Model Quantization

It is about approximating the data by introducing limits on the precision (e.g., rounding errors) and range of a value

- Commonly used method for transforming models for different architectures and reducing computational overhead (e.g., to run inference at the Edge)
- In the context of neural networks, quantization reduces the bits needed to store tensors (i.e., weights, activations, etc.),

Example: mapping 32-bit floating-point numbers (default) to 8-bit integers.

Quantization: The Good & The Bad

Benefits:

- Size reduction
- Reduces computational cost
- Latency reduction
- Better power efficiency

Drawbacks:

- Lower precision/accuracy
- Difficult to predict accuracy ahead of time

Quantization Methods

Pre-training quantization (Quantization aware training)

Induce the expected errors from reduced precision so the network learns the effect of quantization during training

Post-training quantization:

- Weights
- Weights and activations

Assignment 1: Specifications

Assignment 1: Efficient Vision Language Models (VLMs)

- **Aim:** Explore accuracy–efficiency trade-offs in Vision–Language Models (VLMs) by applying post-training quantization.
- Analyze accuracy, latency, memory, and energy usage across different model precisions.
- **Deliverable:** Short report (4–6 pages) + reproducible code & configuration.

Tracks

Choose **one** primary track (**Inference only**, no training) and complete all required tasks. Stretch tasks are optional extra credit

1. **VQA / Multimodal Reasoning Goal**: Ask and answer natural-language questions about an image (zero-shot and few-shot). **Suggested datasets**: VQAv2 (subset), TextVQA (subset), VizWiz (subset).
2. **Image Captioning Goal**: Generate descriptive captions. **Suggested datasets**: COCO 2017 val (5k), NoCaps (subset).

You may propose another VLM task (e.g., ChartQA, DocVQA, OCR-heavy tasks) with instructor approval.

Models (pick one base VLM)

Choose an open-source model that runs on a single consumer GPU or Colab T4. Examples that work with Transformers and common quantizers:

- LLaVA-1.5 (7B/13B variants)
- Qwen2-VL (2B/7B)
- SmolVLM
- Phi-3.5
- Bunny

Tip: Prefer 2B–8B backbones when using free Colab. Larger models are allowed if you have local hardware.

Resources & Compute

- Your own machine (e.g., laptop with recent NVIDIA GPU) or Google Colab (T4)
- Python, PyTorch, Transformers, bitsandbytes; optional: AutoAWQ/GPTQ, llama.cpp/gguf
- Any other platform you like (e.g., Hugging Face Spaces) is OK

What to Measure

For each experiment record:

- **Quality**: task metric(s) below
- **Efficiency**: peak VRAM (MiB), latency per image-prompt, throughput (img/s), and model size on disk
- **Energy (bonus)**: estimate with codecarbon or nvidia-smi logging

Metrics by track

- VQA: overall accuracy on the chosen split; also report random 25-sample human spot-check
- Captioning: CIDEr (primary) and BLEU-4/SPICE if available

Required Experiments

Run the following three configurations on the same data split and seed:

1. Baseline FP16
   a. Load the chosen VLM in FP16 (or BF16) and evaluate.
2. Weight-only 8-bit
   a. Quantize attention/MLP weights to INT8 using bitsandbytes LLM.int8() or AutoAWQ 8-bit.
   b. Keep the vision encoder (vision tower) in FP16 unless your tool supports it.
3. Quantize the vision tower (e.g., ViT) to INT8 using ONNX/TensorRT or bitsandbytes linear8; compare accuracy drop.

Data Protocol

- Use the official validation/dev splits where possible. If you subset, state your indices.
- Limit each image to max side $\leq$ 1024 px for fairness unless your model requires otherwise.

Deliverables

1. **Reproducible code** (link) with README and an experiment config (YAML/JSON) describing: model, quantizer, precision per module, dataset split, seed, and hyper-params.
2. **Short paper (4–6 pages)** using the provided template, containing:
    a. Abstract and Introduction (problem + why efficiency matters)
    b. Methods (model, quantization details, data, training regime)
    c. Results (tables + plots for quality/latency/memory/size; 2–3 qualitative examples with images + predictions)
    d. Discussion (where performance drops, ablations, limitations, and ethical risks such as hallucinations or sensitive content)
    e. Conclusion (insights and recommendations)

3. **One-page appendix:** hardware specs, exact package versions, seeds.

Grading (100 pts)

- Correctness & Repro (20) – code runs from clean env; seeds/splits fixed
- Experimental Rigor (25) – all required configs; fair comparisons; ablations
- Analysis & Writing (25) – clear tables/plots; insightful discussion
- Efficiency Wins (15) – real improvements in latency/VRAM/size vs. FP16
- Polish (15) – clean repo, README, and visual examples

Academic integrity & safety

- Use only open-licensed checkpoints and datasets; cite all sources.
- Inspect outputs for safety issues (privacy, stereotypes, toxic or unsafe text).
- If your model can read text from images (OCR), avoid personal/identifying info.

Assignment 1: Info & Deadlines

- Submission
  - Report (PDF file)
  - Presentation file (PDF file)
  - Source code and artifacts created during the assignment (compressed in a ZIP archive)
- Dates and Timeline
  - Group formation deadline: 25.10.2025
  - Submission deadline: 11.11.2025
  - Presentation: 12.11.2025 (probably only online, further info will be posted)
  - Presentation duration: 10 minutes per group

Please use the forum for assignment-related questions!