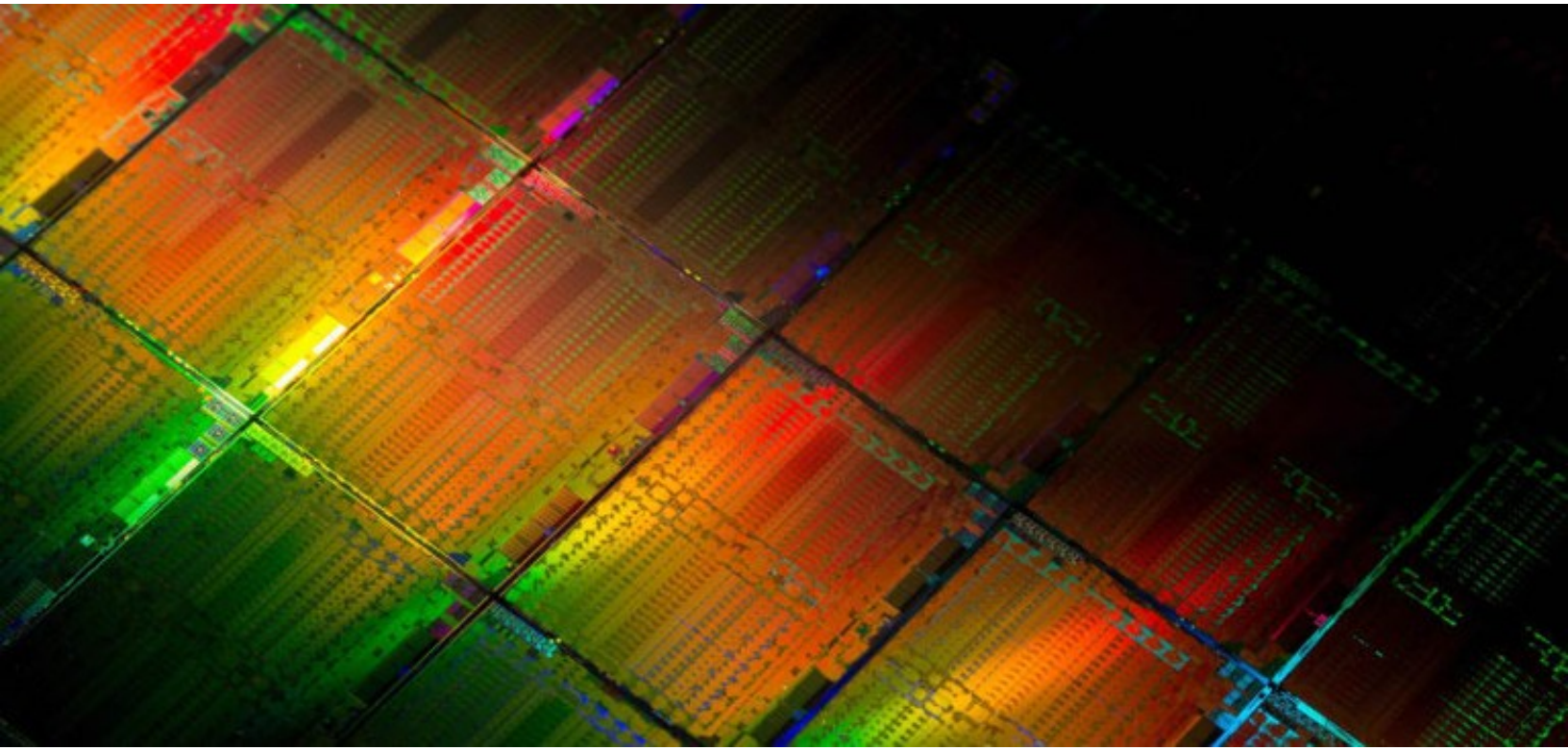


Transformative AI and Compute

Transformative AI and Compute [Summary]

Cross-posted here on the EA Forum.

This is the summary of the series *Transformative AI and Compute - A holistic approach*. You can find the sequence here and the links to the posts below:

1. Compute [1/4]
2. Forecasting Compute [2/4]
3. Compute Governance and Conclusions [3/4]
4. Compute Research Questions and Metrics [4/4]

0. Executive Summary

This series attempts to:

1. Introduce a simplified model of computing which serves as a foundational concept (Part 1 - Section 1).
2. Discuss the role of compute for AI systems (Part 1 - Section 2).
    - In Part 1 - Section 2.3 you can find the updated compute plot you have been coming for.
3. Explore the connection of compute trends and more capable AI systems over time (Part 1 - Section 3).
4. Discuss the compute component in forecasting efforts on transformative AI timelines (Part 2 - Section 4)
5. Propose ideas for better compute forecasts (Part 2 - Section 5).
6. Briefly outline the relevance of compute for AI Governance (Part 3 - Section 6).
7. Conclude this report and discuss next steps (Section 7).
8. Provide a list of connected research questions (Appendix A).
9. Present common compute metrics and discusses their caveats (Appendix B).
10. Provide a list of Startups in the AI Hardware domain (Appendix C).

Abstract

Modern progress in AI systems has been driven and enabled mainly by acquiring more computational resources. AI systems rely on computation-intensive training runs — they require massive amounts of *compute*.

Learning about the compute requirements for training existing AI systems and their capabilities allows us to get a more nuanced understanding and take appropriate action within the technical and governance domain to enable a safe development of potential transformative AI systems.

To understand the role of compute, I decided to (a) do a literature review, (b) update existing work with new data, (c) investigate the role of compute for timelines, and lastly, (d) explore concepts to enhance our analysis and forecasting efforts.

In this piece, I present a brief analysis of AI systems' compute requirements and capabilities, explore compute's role for transformative AI timelines, and lastly, discuss the compute governance domain.

I find that compute, next to data and algorithmic innovation, is a crucial contributor to the recent performance of AI systems. We identify a doubling time of 6.2 months for the compute requirements of the final training run of state-of-the-art AI systems from 2012 to the present.
Next to more powerful hardware components, the spending on AI systems and the algorithmic innovation are other factors that inform the amount of effective compute available — which itself is a component for forecasting models on transformative AI.

Therefore, as compute is a significant component and driver of AI systems' capabilities, understanding the developments of the past and forecasting future results is essential. Compared to the other components, the quantifiable nature of compute makes it an exciting aspect for forecasting efforts and the safe development of AI systems.

I consequently recommend additional investigations in highlighted components of compute, especially AI hardware. As compute forecasting and regulations require an in-depth understanding of hardware, hardware spending, the semiconductor industry, and much more, we recommend an interdisciplinary effort to inform compute trends interpretations and forecasts. Those insights can then be used to inform policymaking, and potentially regulate access to compute.

Epistemic Status

This article is *Exploratory* to *My Best Guess*. I've spent roughly 300 hours researching this piece and writing it up. I am not claiming completeness for any enumerations. Most lists are the result of things I learned *on the way* and then tried to categorize.

I have a background in Electrical Engineering with an emphasis on Computer Engineering and have done research in the field of ML optimizations for resource-constrained devices — working on the intersection of ML deployments and hardware optimization. I am more confident in my view on hardware engineering than in the macro interpretation of those trends for AI progress and timelines.

This piece was a research trial to test my prioritization, interest, and fit for this topic. Instead of focusing on a single narrow question, this paper and research trial turned out to be *more broad* — therefore *a holistic approach*. In the future, I'm planning to work more focused on a narrow relevant research question within this domain. Please reach out.

Views and mistakes are solely my own.

Highlights per Section

1. Compute

- Computation is the manipulation of information.
- There are various types of computation. This piece is concerned with today's predominant type: digital computers.

- With compute, we usually refer to a *quantity of operations* used — computed by our computer/processor. It is also used to refer to computational power, a rate of compute operations per time period.
- Computation can be divided into memory, interconnect, and logic. Each of these components is relevant for the performance of the other.
- For an introduction to integrated circuits/chips, I recommend "AI Chips: What They Are and Why They Matter" (Khan 2020).

2. Compute in AI Systems

- Compute is required for training AI systems. Training requires significantly more compute than inference (the process of using a trained model).
- Next to algorithmic innovation and available training data, available compute for the final training run is one of the significant drivers of more capable AI systems.
- OpenAI observed in 2018 that since 2012 the amount of compute used in the largest AI training runs has been doubling every 3.4 months.
- In our updated analysis (n=57, 1957 to 2021), we observe a doubling time of 6.2 months between 2012 and mid-2021 (n=45).

3. Compute and AI Alignment

- Compute is a component (an input) of AI systems that led to the increasing capabilities of modern AI systems. There are reasons to believe that progress in computing capabilities, independent of further progress in algorithmic innovation, might be sufficient to lead to a transformative AI[1].
- Compute is a fairly coherent and quantifiable feature — probably the easiest input to AI progress to make reasonable quantitative estimates of.
    - Measuring the quality of the other inputs, data, and algorithmic innovation, is more complex.
    - Strongly simplified, compute only consists of one input axis: more or less compute — where we can expect that an increase in compute leads to more capable and potentially unsafe systems.
- According to "The Bitter Lesson", progress arrives via approaches based on scaling computation by search and learning, and not by building knowledge into the systems. Human intuition has been outpaced by more computational resources.
- The strong scaling hypothesis is stating that we only need to *scale* a specific architecture, to achieve transformative or superhuman capabilities — this architecture might already be available. Scaling an architecture implies *more compute* for the training run.
- Cotra's report on biological anchors forecasts the computational power/effective compute required to train systems that resemble, e.g. the human brain's performance. Those estimates can provide compute milestones for transformative capabilities of AI systems.

4. Forecasting Compute

- For transformative AI timeline models with compute milestones, we are interested in how much effective compute we have available at year *Y*.
    - We can break this down into (1) **compute costs**, (2) **compute spending**, and (3) **algorithmic progress**.

- **Hardware progress**: For forecasting hardware progress, no single model can explain the improvements of the last years. Instead, a mix of Moore's Law, chip architectures, and hardware paradigms are applicable models and categories to think about progress.
    - Performance improvements can happen significantly faster than the pure improvement in transistor count and density (Moore's law) would indicate.
    - We will see a fragmentation of applications into the *slow* and *fast lane.* High-demand applications will move to the *fast lane* by designing and benefitting from specialized processors. In contrast, low-demand applications will be stuck in the *slow lane* running on general-purpose processors. We should assume that AI will be on the *fast lane*.
- **Economy of scale**: There will *either* be room for improvement in chip design, or chip design will *stabilize* which enables an economy of scale. Our hardware will *first get better* and then *get cheaper.*
- **Hardware spending**: The current increase in spending is not sustainable; however, it could still significantly increase with estimates up to 1% of (US) GDP (a megaproject, like the Manhattan Project or the Apollo program).
    - However, it is still unclear which percentage of the compute trend has been due to increased spending or to more performant hardware.

5. Better Compute Forecasts

- Researchers should share insights into their AI system's training by disclosing the amount of compute used and its connected details.
    - Ideally, we should make it a requirement for publications and reduce the technical burden of recording the used compute.
- For forecasting hardware progress, we can rely on conceptual models and categories of innovation. We can break this down into:
    - Progress in current computing paradigms
    - Economy of scale for existing technologies
    - Introduction of new computing paradigms
    - Unknown unknowns
- We should also monitor the dominant design strategy as this informs our forecasts. The three design strategies are (1) hardware-driven algorithm design, (2) algorithm-driven hardware design, and (3) co-develop hardware and algorithm.
- Metrics, such as FLOPS/$, often give limited insights, as they only represent one of the three computer components (memory, interconnect, and logic). Understanding their limitations is essential for forecasting.

6. Compute Governance

- Compute is a unique AI governance node due to the required physical space, energy demand, and the concentrated supply chain. Those features make it a governable candidate.
- Controlling and governing access to compute can be harnessed to achieve better AI safety outcomes, for instance restricting compute access to non-safety-aligned actors.
- As compute becomes a dominant factor of costs at the frontier of AI research, it may start to resemble high-energy physics research, where a significant amount of the budget is spent on infrastructure (unlike previous trends of CS research where the equipment costs have been fairly low).

7. Conclusions

- In terms of published papers, the research on compute trends, compute spending, and algorithmic efficiency (the field of macro ML research) is minor and more work on this intersection could quickly improve our understanding.
- The field is currently bottlenecked by available data on macro ML trends: total compute used to train a model is rarely published, nor is spending. With these, it would be easier to estimate algorithmic efficiency and build better forecasting models.
- The importance of compute also highlights the need for ML engineers working on AI safety to be able to deploy gigantic models.
    - Therefore, more people should consider becoming an AI hardware expert or working as an ML engineer at safety-aligned organizations and enabling their deployment success.
- But also working on the intersection of technology and economics is relevant to inform spending and understanding of macro trends.
- Research results in all of the mentioned fields could then be used to inform compute governance.

Acknowledgments

This work was supported and conducted as a summer fellowship at the Stanford Existential Risks Initiative (SERI). Their support is gratefully acknowledged. I am thankful for joining this program and would like to thank the organizers for enabling this, and the other fellows for the insightful discussions.

I am incredibly grateful to Ashwin Acharya and Michael Andregg for their mentoring throughout the project. Michaels thoughts on AI hardware nudged me to reconsider my current research interest and learn more about AI and compute. Ashwin for bouncing off ideas, the wealth of expertise in the domain, and helping me put things into the proper context. Thanks for the input! I was looking forward to every meeting and the thought-provoking discussions.

Thanks to the Swiss Existential Risk Initiative (CHERI) for providing the social infrastructure during my project. Having the opportunity to organize such an initiative in a fantastic team and being accompanied by motivated young researchers is astonishing.

I would like to express my thanks to Jaime Sevilla, Charlie Giattino, Will Hunt, Markus Anderljung, and Christopher Phenicie for your input and discussing ideas.

Thanks to Jaime Sevilla, Jeffrey Ohl, Christopher Phenicie, Aaron Gertler, and Kwan Yee Ng for providing feedback on this piece.

References

- Ahmed, Nur, and Muntasir Wahed. 2020. “The De-Democratization of AI: Deep Learning and the Compute Divide in Artificial Intelligence Research.” *ArXiv:2010.15581 [Cs]*, October. http://arxiv.org/abs/2010.15581.

- Amodei, Dario, and Danny Hernandez. 2018. “AI and Compute.” OpenAI. May 15, 2018. https://openai.com/blog/ai-and-compute/.
- Anderljung, Markus, and Alexis Carlier. 2021. “Some AI Governance Research Ideas.” Some AI Governance Research Ideas - EA Forum. March 6, 2021. https://forum.effectivealtruism.org/posts/kvkv6779jk6edygug/some-ai-governance-research-ideas.
- Branwen, Gwern. 2020. “The Scaling Hypothesis,” May. https://www.gwern.net/Scaling-hypothesis.
- Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. “Language Models Are Few-Shot Learners.” *ArXiv:2005.14165 [Cs]*, July. http://arxiv.org/abs/2005.14165.
- Carey, Ryan. 2018. “Interpreting AI Compute Trends.” AI Impacts. July 10, 2018. https://aiimpacts.org/interpreting-ai-compute-trends/.
- Carlsmith, Joseph. 2020. “How Much Computational Power Does It Take to Match the Human Brain?” Open Philanthropy. August 14, 2020. https://www.openphilanthropy.org/brain-computation-report.
- Centre for the Governance of AI. 2020. “A Guide to Writing the NeurIPS Impact Statement.” *Medium* (blog). May 19, 2020. https://medium.com/@GovAI/a-guide-to-writing-the-neurips-impact-statement-4293b723f832.
- Cotra, Ajeya. 2020. “Draft Report on AI Timelines.” September 19, 2020. https://www.alignmentforum.org/posts/KrJfoZzpSDpnrv9va/draft-report-on-ai-timelines.
- Crox, John. 2019. “On AI and Compute - EA Forum.” EA Forum. March 4, 2019. https://forum.effectivealtruism.org/posts/8wEDjvpcdACvYGQTq/on-ai-and-compute.
- Dafoe, Allan. 2018. “AI Governance: A Research Agenda.” Centre for the Governance of AI, Future of Humanity Institute, University of Oxford. https://www.fhi.ox.ac.uk/wp-content/uploads/GovAI-Agenda.pdf.
- Davidson, Tom. 2021. “Report on Semi-Informative Priors.” Open Philanthropy. March 25, 2021. https://www.openphilanthropy.org/blog/report-semi-informative-priors.
- Finnveden, Lukas. 2020. “Extrapolating GPT-N Performance - AI Alignment Forum.” December 18, 2020. https://www.alignmentforum.org/posts/k2SNji3jXaLGhBeYP/extrapolating-gpt-n-performance.
- Garfinkel, Ben. 2018. “Reinterpreting ‘AI and Compute.’” AI Impacts. December 18, 2018. https://aiimpacts.org/reinterpreting-ai-and-compute/.
- Grace, Katja, John Salvatier, Allan Dafoe, Baobao Zhang, and Owain Evans. 2016. “2016 Expert Survey on Progress in AI.” *AI Impacts* (blog). December 14, 2016. https://aiimpacts.org/2016-expert-survey-on-progress-in-ai/.

- Hernandez, Danny, and Tom B. Brown. 2020. “Measuring the Algorithmic Efficiency of Neural Networks.” *ArXiv:2005.04305 [Cs, Stat]*, May. http://arxiv.org/abs/2005.04305.
- Hestness, Joel, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. 2017. “Deep Learning Scaling Is Predictable, Empirically.” *ArXiv:1712.00409 [Cs, Stat]*, December. http://arxiv.org/abs/1712.00409.
- Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. “Scaling Laws for Neural Language Models.” *ArXiv:2001.08361 [Cs, Stat]*, January. http://arxiv.org/abs/2001.08361.
- Khan, Saif M. 2020. “AI Chips: What They Are and Why They Matter.” *Center for Security and Emerging Technology* (blog). April 2020. https://cset.georgetown.edu/research/ai-chips-what-they-are-and-why-they-matter/.
- ———. 2021. “The Semiconductor Supply Chain.” *Center for Security and Emerging Technology* (blog). January 2021. https://cset.georgetown.edu/publication/the-semiconductor-supply-chain/.
- Los Alamos National Laboratory. 2013. “Massive Infrastructures Are Needed to Support Supercomputers.” March 25, 2013. https://www.lanl.gov/discover/publications/national-security-science/2013-april/what-is-under the floor-of-a-supercomputer.php.
- Lyzhov, Alex. 2021. “‘AI and Compute’ Trend Isn’t Predictive of What Is Happening.” “AI and Compute” Trend Isn’t Predictive of What Is Happening - AI Alignment Forum. February 4, 2021. https://www.alignmentforum.org/posts/wfpdejMWog4vEDLDg/ai-and-compute-trend-isn-t-predictive-of-what-is-happening.
- Microsoft Documentation. 2020. “Deploy ML Models to FPGAs - Azure Machine Learning.” September 24, 2020. https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-fpga-web-service.
- Mirhoseini, Azalia, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, et al. 2021. “A Graph Placement Methodology for Fast Chip Design.” *Nature* 594 (7862): 207–12. https://doi.org/10.1038/s41586-021-03544-w.
- MLCommons. 2021. “MLCommons^{TM} Releases MLPerf^{TM} Training v1.0 Results.” MLCommons. June 30, 2021. https://mlcommons.org/.
- Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. “Language Models Are Unsupervised Multitask Learners,” February, 24.
- Sevilla, Jaime, Pablo Villalobos, Juan Felipe Cerón, Matthew Burtell, and Lennart Heim. 2021. “Parameter, Compute and Data Trends in Machine Learning.” *Google Sheets* (blog). June 19, 2021.

https://docs.google.com/spreadsheets/d/1AAIebjNsnJj_uKALHbXNfn3_YsT6sHXtCU0q7OIPuc4/.

- Shalf, John. 2020a. “The Future of Computing beyond Moores Law.” *Philosophical Transactions of the Royal Society A*, March. https://doi.org/10.1098/rsta.2019.0061.
- ———. 2020b. “Computing Beyond Moore’s Law.” July 14. https://cs.lbl.gov/assets/CSSSP-Slides/20200714-Shalf.pdf.
- Sutton, Rich. 2019. “The Bitter Lesson.” March 13, 2019. http://www.incompleteideas.net/IncIdeas/BitterLesson.html.
- Thompson, Neil C., Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. 2020. “The Computational Limits of Deep Learning.” *ArXiv:2007.05558 [Cs, Stat]*, July. http://arxiv.org/abs/2007.05558.
- Thompson, Neil C., and Svenja Spanuth. 2021. “The Decline of Computers as a General Purpose Technology.” *Communications of the ACM* 64 (3): 64–72. https://doi.org/10.1145/3430936.

---

1. Transformative AI, as defined by Open Philanthropy in this blogpost: “*Roughly and conceptually, transformative AI is AI that precipitates a transition comparable to (or more significant than) the agricultural or industrial revolution.*” ↩

What is Compute? - Transformative AI and Compute [1/4]

Cross-posted here on the EA Forum.

Transformative AI and Compute - A holistic approach - Part 1 out of 4

This is part one of the series *Transformative AI and Compute - A holistic approach*. You can find the sequence here and the summary here.

This work was conducted as part of Stanford's Existential Risks Initiative (SERI) at the Center for International Security and Cooperation, Stanford University. Mentored by Ashwin Acharya (Center for Security and Emerging Technology (CSET)) and Michael Andregg (Fathom Radiant).

This post attempts to:

1. Introduce a simplified model of computing which serves as a foundational concept (Section 1).
2. Discuss the role of compute for AI systems (Section 2).
    - In Section 2.3 you can find the updated compute plot you have been coming for.
3. Explore the connection of compute trends and more capable AI systems over time (Section 3).

Epistemic Status

This article is *Exploratory* to *My Best Guess*. I've spent roughly 300 hours researching this piece and writing it up. I am not claiming completeness for any enumerations. Most lists are the result of things I learned *on the way* and then tried to categorize.

I have a background in Electrical Engineering with an emphasis on Computer Engineering and have done research in the field of ML optimizations for resource-constrained devices — working on the intersection of ML deployments and hardware optimization. I am more confident in my view on hardware engineering than in the macro interpretation of those trends for AI progress and timelines.

This piece was a research trial to test my prioritization, interest and fit for this topic. Instead of focusing on a single narrow question, this paper and research trial turned out to be *more broad* — therefore *a holistic approach*. In the future, I'm planning to work more focused on a narrow relevant research questions within this domain. Please reach out.

Views and mistakes are solely my own.

1. Compute

Highlights

- Computation is the manipulation of information.

- There are various types of computation. This piece is concerned with today's predominant type: digital computers.
- With compute, we usually refer to a *quantity of operations* used — computed by our computer/processor. It is also used to refer to computational power, a rate of compute operations per time period.
- Computation can be divided into **memory**, **interconnect**, and **logic**. Each of these components is relevant for the performance of the other.
- For an introduction to integrated circuits/chips, I recommend "AI Chips: What They Are and Why They Matter" (Khan 2020).

This section discusses the term compute and introduces one conceptual model of a processor — a helpful analogy for analyzing trends in compute.

Compute is the manipulation of information or any type of calculation — involving arithmetical and non-arithmetical steps. It can be seen as happening within a closed system: a *computer*. Examples of such physical systems include digital computers, analog computers, mechanical computers, quantum computers, or wetware computers (your brain).[1]

For this report, I am focusing on today's predominant type of computing: digital computing. This is for two reasons: First, it is the predominant type of computing of human engineered systems. Second, my expertise lies within the digital computing domain.

That does not mean that other computing paradigms, such as quantum computing[2], are not of relevance. They require a different analysis and are out of scope for this piece. I believe these paradigms are unlikely to be important to AI progress in the next 10 years, as they are far less mature than digital computing. Beyond that point, however, they might become cost-competitive with digital computing, and I am uncertain as to whether they could speed up AI progress by a significant amount.

1.1 Logic, Memory and Interconnect

For this piece, we can refer to compute as either arithmetical or non-arithmetical operations. An example of an arithmetic operation is adding two numbers (`a ← b + c`) or multiplying two numbers (`a ← b x c`), whereas non-arithmetic operations are, for example, comparisons: "*Is a greater than b* (`a > b`)*?*" Based on the resolution, we then continue with a selection of arithmetic operations. Such a selection of operations is called a program or application.

Therefore, when referring to *compute used*, we simply count how many of those basic operations the computer has done. *Compute used* is a quantity of operations. A floating point operation (FLOP) is such an example of an operation on floating point numbers. Those metrics and their nuances are discussed in Part 4 - Appendix B.

For an application, such as calculating the quickest route from *A* to *B*, we require a clearly defined number of operations, e.g., 10,000,000 of such operations. This number does not give any insights into *how long* it takes to compute this. Think of it as a distance; how long it will take depends on your vehicle and many other external factors.

Now that we have defined the compute consists of several operations, we need something which executes those operations: a computer.

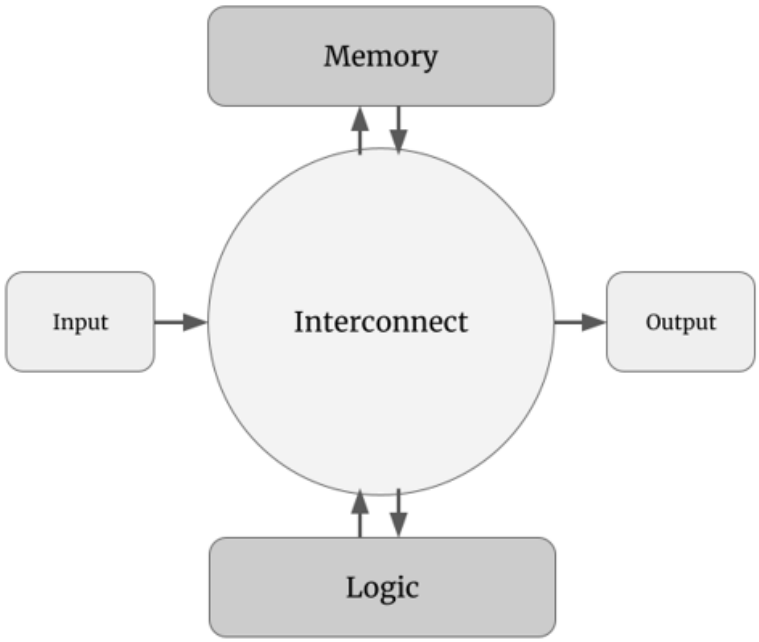


Figure 1.1: A conceptual model of a computer consisting of memory, interconnect, and logic.

The three essential elements of a computer are: Logic, Memory, Interconnect.

- **The logic** component is where we conduct the arithmetical and non-arithmetical operations.
- **The memory** supplies the logic via the interconnect with the information *on which* and which the logic processes. It is also the place where we store our results.
- **The interconnect** supplies the logic and memory with the information that *needs to be processed* or *just got processed.*

Whereas many people often only think about the logic unit as part of a computer, the memory and interconnect are as important. We need to transport the manipulated information and store it.

Consequently, all of those three components are influencing one another:

- If our memory capacity is limited, we cannot save all of the results.
- If the interconnect does not supply the logic fast enough with more information, the logic is *waiting* and *stalled*.

This is a basic model which applies to a wide variety of computers. Your brain can also be conceptualized this way. There are places where you store information: *memory; we* can recall this information (if lucky): *interconnect*, then manipulate it: *logic,* and potentially save it again via the *interconnect*. Depending on your task, you might be *bottlenecked* by either one of those components. You might not process as much information as coming in. Your *interconnect* is faster than your *logic* — you are *overutilized* (>100%).

In contrast, modern computers are often *underutilized* (<100%): The logic can theoretically process more information per second than the interconnect supplies:

$P_{Logic} > BW_{Interconnect}$.

Consequently, solely improving one component of our model does not lead to better performance. In general, during the last decades, we have seen innovations that fall into either one of those three buckets to increase performance:

1. Increasing the processing speed of the logic: more operations per second [*OP per second*].
2. Increasing the memory capacity: we can recall and save more information that needs to be or has been processed [*Unit of information*].
3. Increasing the interconnect speed: the amount of information we can supply to our logic or memory per second [*Unit of information per second*]:

Examples of such innovations are pipelining, multicore, heterogeneous architectures, or specialized processors. In the future, I plan to publish a more extended introduction to compute — understanding those innovations helps us to understand caveats of performance metrics and forecast their trends.

1.2 Chips or Integrated Circuits

Our theoretical model for a computer then gets embedded into a substrate we call a chip or an integrated circuit.

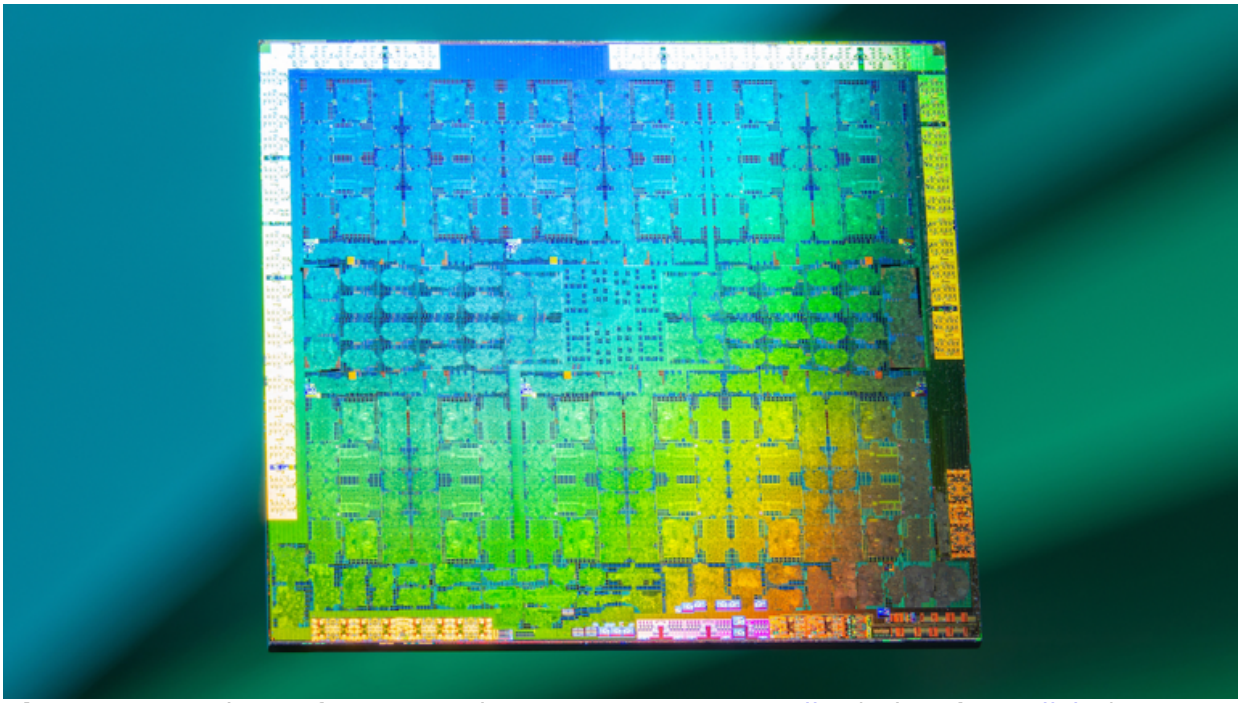


Figure 1.2: A photo of NVIDIA Turing GeForce RTX 2080 die. (Taken from Flickr.)

I would consider this the most complex machine we humans have ever created. I will not get into the details on how we get our theoretical computer model onto an actual working chip. For an introduction to chips, especially AI chips, I can recommend the piece: "AI Chips: What They Are and Why They Matter" (Khan 2020).

2. Compute in AI Systems

Highlights

- Compute is required for training AI systems. Training requires significantly more compute than inference (the process of using a trained model).
- Next to algorithmic innovation and available training data, available compute for the final training run is a significant driver of more capable AI systems.
- OpenAI observed in 2018 that since 2012 the amount of compute used in the largest AI training runs has been doubling every 3.4 months.
- In our updated analysis (n=57, 1957 to 2021), we observe a doubling time of 6.2 months between 2012 and mid-2021.

Computation has two obvious roles in AI systems: training and inference. This section discusses those and presents compute trends.

2.1 Computing in AI Systems

Training

Training a neural network, also referred to as learning, describes the process of determining the value of the weights and biases in the network. It is a fundamental component of ML systems, be it supervised, unsupervised or reinforcement learning.[3]

When we refer to compute used for training, we usually refer to the amount of compute used for the final training run. Common metrics are operations (OPs), floating point operations (FLOPs) or Petaflop/s-days[4]. Once we have trained an AI system, we theoretically can give a quantity of operations conducted — this is the metric we are interested in.

Nonetheless, during the development stage, we also require training runs for trial and error where we tweak the architecture, hyper-parameters and others. Consequently, the amount of compute for the final training run might be useful as a proxy for the capabilities of the AI system, however, the total cost can differ significantly.

Inference

Once a network is trained, the output can be computed by running the computation, determined by the weights and network architecture. This process is referred to as *inference* or *forward pass*. A trained network can be distributed and then deployed for applications. At this point, the network is static: all computations and intermediate steps are defined. Only an input is necessary to carry out inference. An inference is done over and over. Examples are doing a single Google search, asking a voice assistant such as Siri or Alexa, and others.

Training is computationally more complex than inference. There are two primary reasons: First, the training of the weights and biases is an iterative process – usually, thousands of backward passes are required for a single input to obtain the desired result. Secondly, for the training process, at least for supervised learning, the labeled training data needs to be available to the computation system, which requires memory

capacity. Therefore, combining those two points, training is a significantly more complex process regarding memory and computational resources, as one requires thousands of backward passes for a single input of the enormous training set.[5]

This also leads to important consequences for the field of compute governance, as once an actor acquires a trained model, the required amount of compute for running the inference is significantly less, which can lead to so-called model theft (Anderljung and Carlier 2021).

We have discussed the two distinct computations of AI systems: training and inference. For this piece, I am mostly concerned about the required compute for the final training run[6], as we have observed meaningful trends for this metric over time, and there seems to be an agreement that the amount of compute used plays a role for the capabilities of the AI systems (Section 3 discusses this connection).

For the rest of the piece, my usage of the term "*compute used*" is synonymous with "*the amount of compute used for the final training run of an AI system*".

2.2 Compute Trends: 2012 to 2018

In the blogpost "AI and Compute", Amodei and Hernandez analyze the compute used for final training runs (Amodei and Hernandez 2018). At this point, it remains the best available analysis.

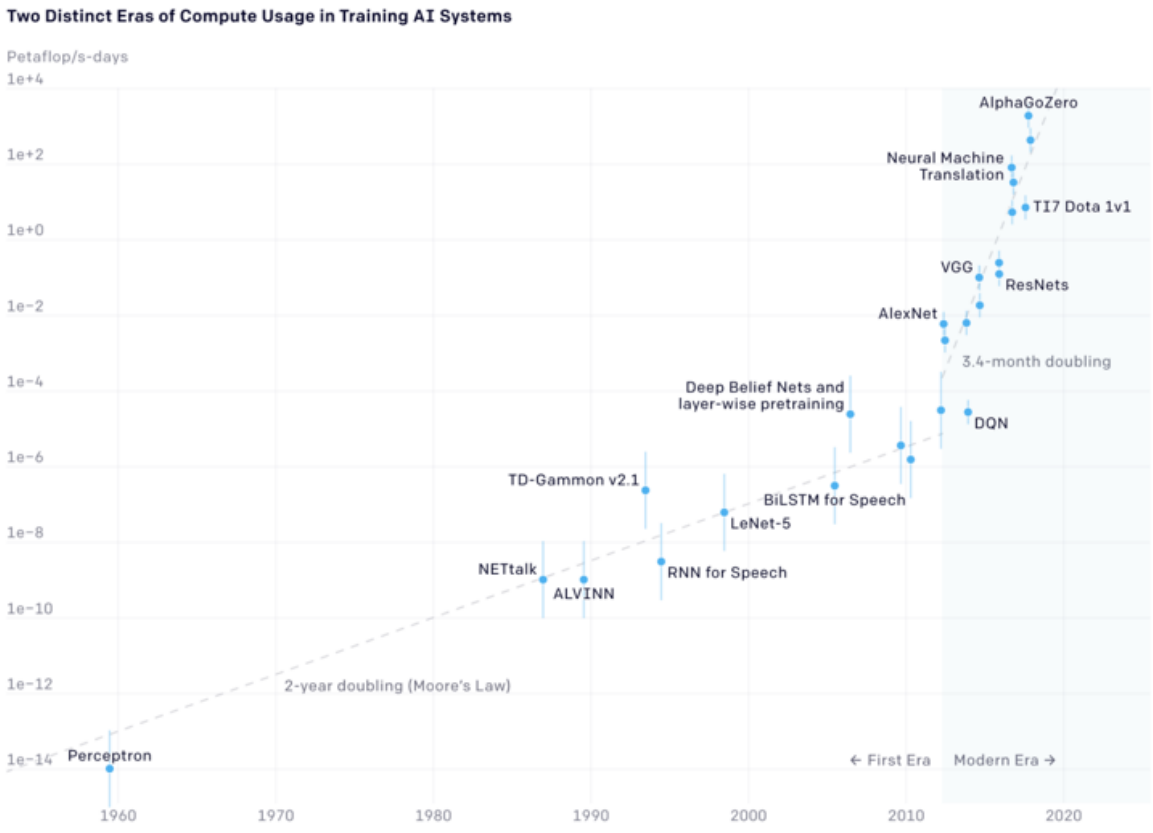


Figure 2.1: Total amount of compute used for final training runs of selected AI

systems. Source: (Amodei and Hernandez 2018).

We can summarize the main insights as follows:

- We can divide the history of compute used for AI systems based on the trend lines into two eras: the first and modern era.
  - The modern era started with AlexNet in 2012. AlexNet was the first publication that leveraged graphical processing units (GPUs) for the training run (we discuss this development in Part 2 - Section 4.2).
- From 2012 to 2018: the amount of compute used in the largest AI training runs has been increasing exponentially, with a **doubling time of 3.4 months**
  - Since 2012 this metric has grown **by more than 300,000x.**

The authors do not comment on the breakdown of this trend: is it dominated by increased spending or hardware improvements? Other people at the time noted that this trend has to be due to increased spending on compute since this growth was far faster than growth in computing capabilities per dollar (Carey 2018; Garfinkel 2018; Crox 2019). Eventually, this would lead to massive spending.

The blogpost closes with an outlook where they could imagine this trend continuing, e.g., due to continued improved hardware by hardware startups.[7] As this piece's latest AI system was AlphaZero which was published in December 2018, we tried to crowdsource more compute estimates and take an updated look.

2.3 Compute Trends: An Update[8]

We are not the first to take an updated look at this trend. Lyzhov sketched this (Lyzhov 2021) and also Branwen discussed this (Branwen 2020). Also, there are several posts that commented on AI and Compute (Carey 2018; Garfinkel 2018; Crox 2019). I will reference some interpretations in Part 2 - Section 4: Forecasting Compute.

Our dataset (Sevilla et al. 2021) contains the same results as the one used in “AI and Compute” and includes a total of 57 models. We have added some recent notable systems such as GPT-3, AlphaFold, PanGu-α, and others. Note that those estimates are either our or others’ *best guesses*. Most publications do not disclose the amount of compute used. You can find the calculations and sources in the note of the corresponding cell. The public dataset is available and continuously updated as its own independent project. We would appreciate any suggestions and comments. An interactive version of the graph below is available here.

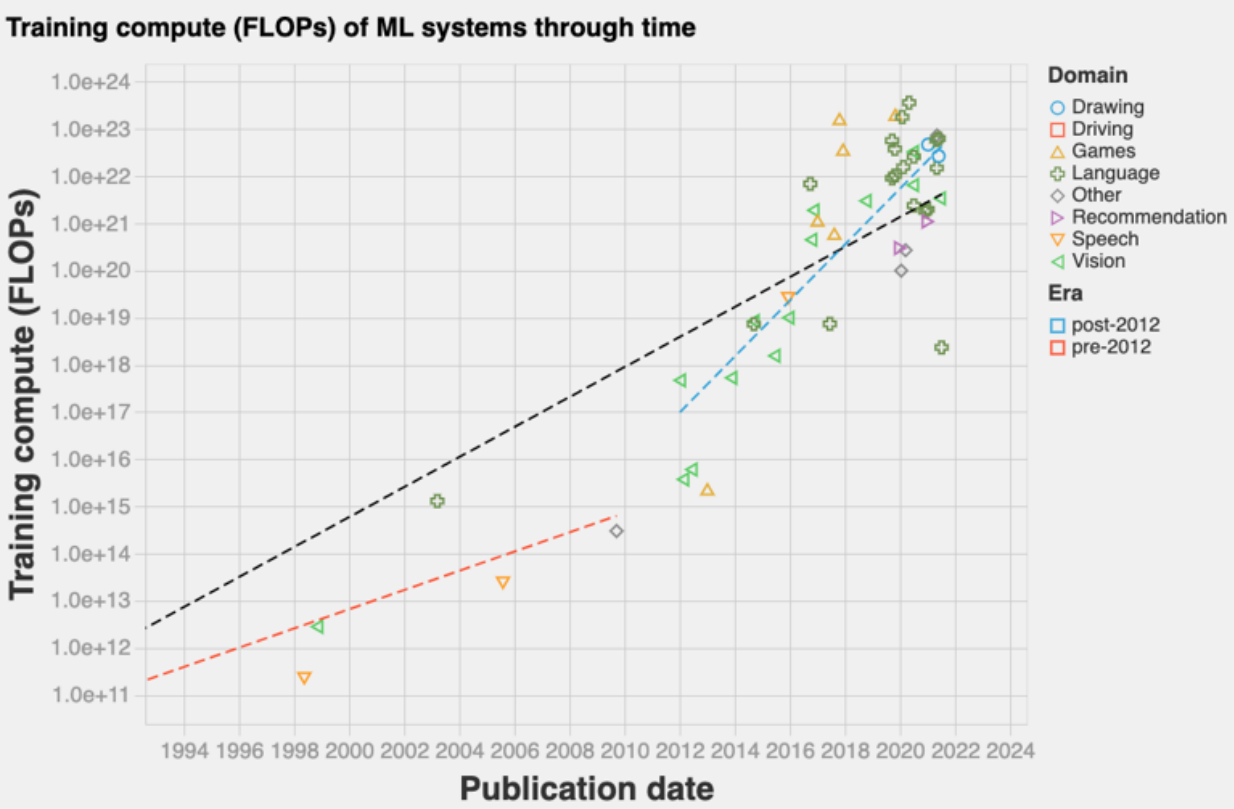


Figure 2.2: Training compute of ML systems through time. An interactive version of the graph is available here and the dataset here.

The key insights are:

- If we analyze the same period, as in the AI and Compute piece, from AlexNet to AlphaZero (12-2017), we observe a doubling time of 3.6 months (versus 3.4 months (Amodei and Hernandez 2018)).
- Our data includes 28 additional models which have been released since then.
- Groundbreaking AI systems, such as AlphaFold, required less compute than previous systems.
- Only three systems used more compute than the leader of the original “AI and Compute”, AlphaGo Zero. Those systems are AlphaStar, Meena, and GPT-3.
- In our dataset, we find the following doubling times:
  - 1957 to 2021 doubling time: **11.5 months**
  - Pre 2012 doubling time: **18.0 months**
  - Post 2012 doubling time: **6.2 months**
- However, note that the results and trends should be taken with caution, as this model relies on an incomplete dataset due to an ambiguous selection process and the compute numbers are mostly estimates.

For now, we do not interpret those trends. I do not want to make strong predictions that new state-of-the-art AI systems will not end up using more compute. Nonetheless, we should also not assume that the incredibly fast doubling rate seen in “AI and Compute” will occur in the future, either. For some discussion around this update, see Lyzhov’s post “"AI and Compute" trend isn't predictive of what is happening“ (Lyzhov 2021).

We also think that the selection of the models in the dataset plays an important role. Our heuristic criteria of inclusion was along the lines of: "*important publication within the field of AI OR lots of citations OR performance record on common benchmark*." Also, it is unclear on which models we should base this trend. The piece AI and Compute also quickly discusses this in the appendix. Given the recent trend of efficient ML models due to emerging fields such as Machine Learning on the Edge, I think it might be worthwhile discussing how to integrate and interpret such models in analyses like this — ignoring them cannot be the answer.

We have discussed the historical trends of compute for final training runs and seen a tremendous increase in compute used — but has this resulted in more capable AI systems? We explore this question in the next section.

3. Compute and AI Alignment

Highlights

- Compute is a component (an input) of AI systems that led to the increasing capabilities of modern AI systems. There are reasons to believe that progress in computing capabilities, independent of further progress in algorithmic innovation, might be sufficient to lead to a transformative AI[9].
- Compute is a fairly coherent and quantifiable feature — probably the easiest input to AI progress to make reasonable quantitative estimates of.
    - Measuring the quality of the other inputs, data and algorithmic innovation, is more complex.
    - Strongly simplified, compute only consists of one input axis: more or less compute — where we can expect that an increase in compute leads to more capable and potentially unsafe systems.
- According to "The Bitter Lesson", progress arrives via approaches based on scaling computation by search and learning, and not by building knowledge into the systems. Human intuition has been outpaced by more computational resources.
- The strong scaling hypothesis is stating that we only need to *scale* a specific architecture, to achieve transformative or superhuman capabilities — this architecture might already be available. Scaling an architecture implies *more compute* for the training run.
- Cotra's report on biological anchors forecasts the computational power/effective compute required to train systems that resemble, e.g. the human brain's performance. Those estimates can provide compute milestones for transformative capabilities of AI systems.

We have seen an enormous upwards trend in the used compute for AI systems. It is also well known that modern AI systems are *more capable* at various tasks. However, the relative importance of three driving factors (algorithmic innovation, data, and compute) for this progress is unclear.

This section explores the role of compute for more capable systems and discusses hypotheses that favor compute to be a predominant factor in the past, and the future. In my opinion, there seems to be consensus amongst experts that compute has played an important role (at least as important as the other two factors) in AI progress over the last decade and will continue doing so.

My best guess is that compute is not an input that we can leverage to create safe AI systems. Strongly simplified, compute only consists of one input axis: *more* or *less* compute — where we can expect that an increase in compute leads to more capable and potentially unsafe systems (as I will discuss in this section). Data and algorithms have more than one input axis which could, in theory, be used to affect the safety of an AI system. Algorithms could be designed to understand human intent and *be aligned*, and data could be more or less biased, leading to wrong interpretations and learnings.

Consequently, within the field of AI governance, compute is a unique sub-domain, and its importance could be harnessed to achieve better AI alignment outcomes, by,. e.g., regulating access to compute. I will discuss this in Part 3 - Section 6.

The following subsections will explore different hypotheses about the role of compute for AI systems capabilities and their safety: “The Bitter Lesson”, the scaling hypothesis, qualitative assessments and compute milestones using biological anchors.

3.1 The Bitter Lesson

Rich Sutton wrote the commonly-cited piece “The Bitter Lesson”. He argues that algorithmic innovation mostly matters insofar as it creates general-purpose systems capable of making use of great amounts of compute. He bases this lesson as a historical observation:

1. AI researchers tried to integrate knowledge into their agents/systems.
2. This has helped in the short-term.
3. However, in the long run, this progress fades and it even *inhibits* further progress.
4. Breakthrough progress eventually then comes by a contrary approach that relies on scaling computation.

It is a *bitter* lesson because the latter approach is more successful than the human-centric approach, which relies on intuition from the researchers and is often *more personally satisfying*. (Sutton 2019):

> “One thing that should be learned from the bitter lesson is the great power of general purpose methods, of methods that continue to scale with increased computation even as the available computation becomes very great. The two methods that seem to scale arbitrarily in this way are *search* and *learning*.”

He closes with:

> “We want AI agents that can discover like we can, not which contain what we have discovered.”

3.2 Scaling Hypothesis

The scaling hypothesis is similar to the *bitter lesson*. Gwern describes it the following (Branwen 2020):

> “The strong *scaling hypothesis* is that, once we find a scalable architecture like self-attention or convolutions, which like the brain can be applied fairly uniformly, we can simply train ever larger NNs and ever more sophisticated behavior will emerge naturally as the easiest way to optimize for all the tasks & data.

> More powerful NNs are ‘just’ scaled-up weak NNs, in much the same way that human brains look much like scaled-up primate brains.”

Consequently, to achieve more capabilities, we only need to *scale* a specific architecture (where it is still open if we already have *this* architecture). Scaling an architecture implies more parameters and requires more compute for training this network. According to Gwern “the scaling hypothesis has only looked more and more plausible every year since 2010” (Branwen 2020).[10]

GPT-2 (Radford et al. 2019) to GPT-3 (Brown et al. 2020) is an example of scaling and is seen as evidence for the scaling hypothesis. Finnveden extrapolates this trend and discusses this in his forum post “Extrapolating GPT-N performance” (Finnveden 2020).

3.3 AI and Efficiency

We have discussed that increased compute has led to more capable AI systems, but how does its relevance compare to the two other main factors: data and algorithmic innovation?

The best available piece investigating algorithmic innovation is “AI and Efficiency” by Hernandez and Brown from OpenAI. This analysis examines the algorithmic improvement of the last years by having a constant benchmark (AlexNet performance on ImageNet) and training modern AI systems until they achieve similar performance. The algorithmic improvement is then seen in the reduced amount of compute to achieve similar performance. Their analysis shows that we have seen a 44x increase in algorithmic progress. This result is more likely to be an underestimate for various reasons — see their paper for a discussion on this (Hernandez and Brown 2020).

Those insights are important to decompose the *effective compute* available (Figure 3.1).

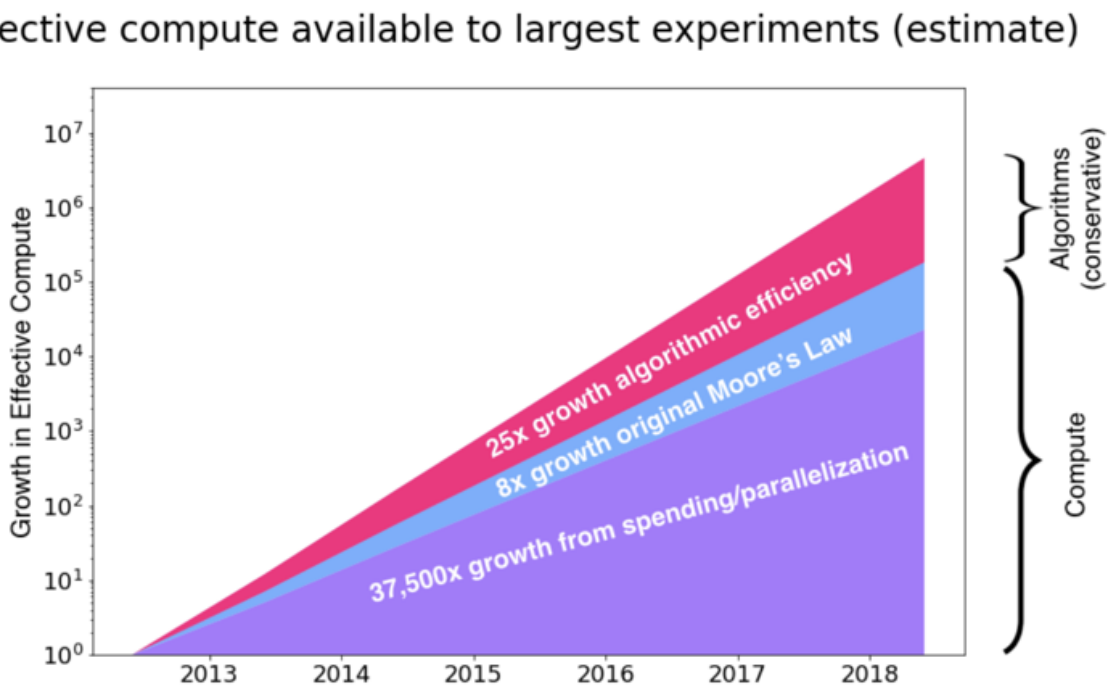


Figure 3.1: Growth in effective compute over time and its decomposition. (Taken from (Hernandez and Brown 2020))

This analysis is highly speculative, according to the authors. Nonetheless, it also highlights our lack of decomposition in the purple part, “37’500x growth from spending/parallelization”.[11]

> “We’re uncertain whether hardware or algorithmic progress actually had a bigger impact on effective compute available to large experiments over this period.”

Consequently, thinking about *effective compute* is a critical concept to measure progress. The idea is also used in the timeline by Cotra, which we will discuss in the next Part 2 Section 4.1.

3.4 Qualitative Assessment

Lastly, in the 2016 Expert Survey on Progress in AI, leading AI researchers were asked about the improvements without the same growth of (1) computing cost, (2) algorithmic progress, (3) training data, (4) research effort, and (5) funding.

For (1) computing cost, the question was the following:

> “Over the last 10 years the cost of computing hardware has fallen by a factor of 20. Imagine instead that the cost of computing hardware had fallen by only a factor of 5 over that time (around half as far on a log scale). How much less progress in AI capabilities would you expect to have seen? e.g. If you think progress is linear in 1/cost, so that 1-5/20=75% less progress would have been made, write ’75’. If you think only 20% less progress would have been made write ’20’.”

For this question, the median answer was 50% and the following distribution:

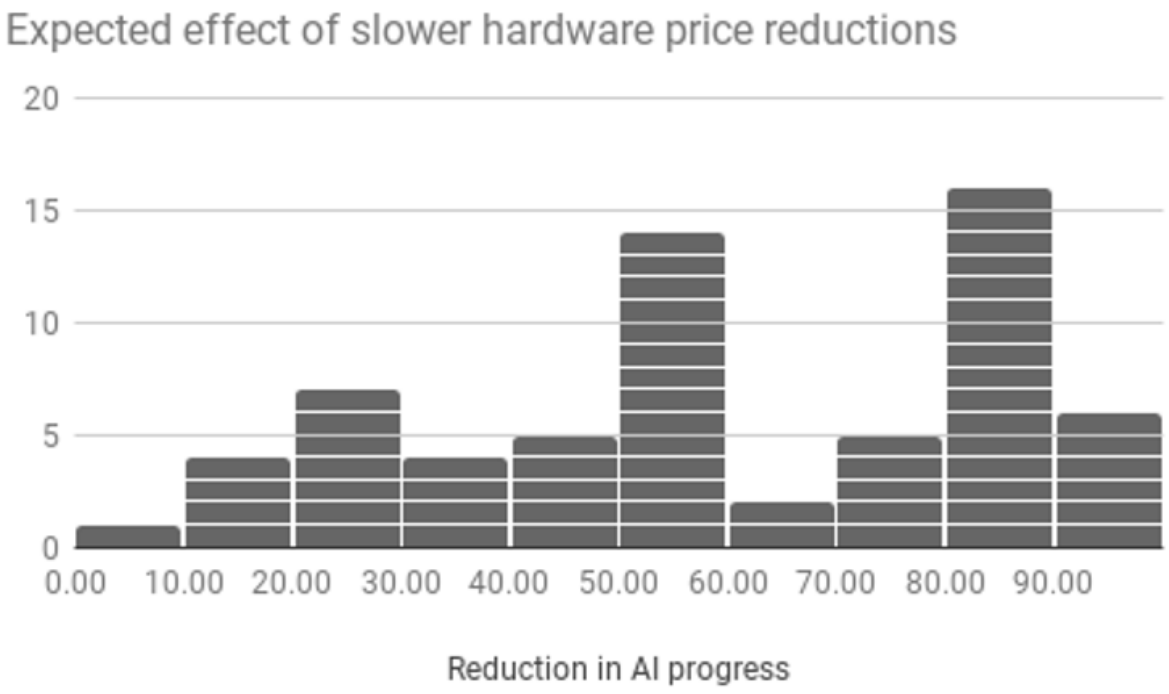


Figure 3.2: Reduction in AI progress if compute costs had only fallen by a factor of 5 instead of 20. (Taken from (Grace et al. 2016).)

The other factors were at most as important or less important than the reduction in hardware prices. For a comparison to the other factors, see the blog post (Grace et al. 2016)**.**[12]

3.5 Compute Milestones

Compute milestones describe estimates and forecasts on the required computational power for matching well-known capable compute systems, such as the human-brain. The theory is that we know computational systems, such as the human-brain, or evolution, which have resulted in *general intelligence.* To this point, we have not created an artificial system yet which breaks even. Consequently, we could be missing some fundamental algorithmic insights and/or our existing AI systems do not match their computational requirement yet.

Once we hit those milestones and no transformative capabilities are unlocked before, we can be fairly sure that the missing pieces are algorithmic. So far it seems like a few algorithmic advances have helped modern AI improve, but a great deal of progress has been due to scaling alone. In some ways, the resultant systems already outperform humans, despite taking far less training and compute to train than, e.g., human evolution. If one accepts the scaling hypothesis, one might believe that using massively more compute will get us to general intelligence, and processes that have led to intelligent behavior in previous systems, such as biological systems, are a natural reference point. For a discussion on this, I refer to the book Superintelligence by Bostrom.

There are various analogies to draw in regards to our components from our basic concept of a computer: logic, interconnect, memory. We are unsure if our capabilities in regards to logic, interconnect and memory within the human brain have been matched, but maybe an individual component has been matched yet.[13] Whereas I'm uncertain about the interconnect *within* the brain, the interconnect *between* different brains is likely to be surpassed. While we as humans are having a hard time interfacing with others due to limited bandwidth communication, such as written text or speech, the interconnect between computer systems seems to have significantly higher bandwidth and does not have the same restrictions as we humans do: connecting and scaling compute systems is easier than it is for human brains.

Various reports introduce these milestones by using biological anchors. Most notably "How Much Computational Power Does It Take to Match the Human Brain?" by (Carlsmith 2020), "Semi-informative priors over AI timelines" by (Davidson 2021), and the draft report on AI timelines by (Cotra 2020). Consequently, those milestones can provide meaningful insights for forecasting transformative AI. We will discuss this in Part 2 - Section 4.1.

Those reports list the following milestones (Davidson 2021; Cotra 2020; Carlsmith 2020):

The amount of compute required to simulate ...

- (1) ... the evolution of the human brain (*Evolutionary hypothesis*)
  - Evolution has created intelligent systems, such as the mammalian brain. This milestone estimates the amount of compute which happened inside the nervous systems throughout evolution.
- (2) ... total computation done over a human lifetime (*Lifetime learning hypothesis*)
  - This milestone is based on the learning process of a human child's brain until they reach the age 30.
- (3) The computational power of the human brain.
  - When could computers perform any cognitive task that the human brain can?
  - "*How Much Computational Power Does It Take to Match the Human Brain?*" (Carlsmith 2020).
- (4) The amount of information in the human genome.
  - This milestone estimates that our system would require as much information in its parameters as there is the human genome, next to being computationally as powerful as the brain.

All of the milestones rely on some kind of estimate on the required amount of (effective) compute.

Therefore, depending on the details of this estimate, matching those biological counterparts could already be unlocked before we reach their computational requirements. We could already have developed more efficient algorithms, will create better algorithms, or other modules of the system, such as memory capacity or the machine-to-machine interconnect, could unlock these milestones earlier. Consequently, those estimates could provide some kind of *upper bound* of when we will reach capabilities similar to the biological counterparts — assuming that it is a pure computational process and we have the *right* algorithms.

3.6 Conclusion

We have learned that compute is one of the key contributors of today's more capable AI systems[14]. Compute is, next to data and algorithmic innovation, important and increased compute has led to more capable AI systems. There are reasons to believe (milestones, and scaling hypothesis) that a continued trend in compute could lead to capable, if not, transformative AI systems. I do not see the list of milestones as complete, rather, those are the ones *I stumbled across*. There are also empirical investigations on scaling laws available (Kaplan et al. 2020; Hestness et al. 2017).

During my limited research time, the scaling hypothesis and "The Bitter Lesson" became more plausible to me. As Sutton describes: It is a *historical observation* — where I would see the recent years as additional confirmation (e.g., GPT-3). I would be interested in the opinions of AI experts on this.

Compute is a quantity, and computational power is a rate, which is clearly defined and measurable — only the stacked layers of abstractions make it not easily *accessible* (discussed in Part 2 - Section 5.1). However, comparing it to the quality of data, or algorithmic innovation, I am more optimistic for compute measurement than for the others. Therefore, compute is an especially interesting component for analyzing the past and the present, but also forecasting the future. Controlling and governing it can be harnessed to achieve better AI alignment outcomes (see Part 3 - Section 6: Compute Governance). The next section explores *forecasting compute*.

Next Post: Forecasting Compute

The next post "*Forecasting Compute [2/4]*" will attempt to:

4. Discuss the compute component in forecasting efforts on transformative AI timelines (Part 2 - Section 4)
5. Propose ideas for better compute forecasts (Part 2 - Section 5).

Acknowledgments

You can find the acknowledgments in the summary.

References

The references are listed in the summary.

---

1. One could argue the universe is a computer as well: pancomputationalism. ↩
2. You can read some thoughts on quantum computing in the series "Forecasting Quantum Computing" by Jaime Sevilla. ↩
3. Compute produces the data as an interactive environment for reinforcement learning. Therefore, more compute leads to more available training data. ↩
4. A petaflop/s is 10^{15} floating point operations per second for one day. A day has $86,400$ seconds $\approx 10^5$ seconds. Therefore, 10^{20} floating point operations. ↩

5. Nonetheless, according to estimates, overall most compute is probably used for the deployed AI systems — inference. Whereas, as outlined, the training process is computational more complex, the repetitive behavior of inference once deployed, leads to overall *more used compute*. In the future those resources could be repurposed for training (if we do not see different hardware for training and inference — discussed in Section 4.2) (*compute for training >> compute for inference* but *number of inferences >> number of training runs*) (Amodei and Hernandez 2018). ↩

6. The final training run refers to the last training of an AI system before stopping updating the learned weights and biases and deploying the network for inference. There are usually dozens to hundreds of training runs of AI systems to tweak the architecture and hyper-parameters optimally. While this metric is relevant for the development costs, it is not an optimal proxy for the systems' capabilities. ↩

7. "*We think it'd be a mistake to be confident this trend won't continue in the short term.*" (Amodei and Hernandez 2018). ↩

8. The data used in this section is coming out of a project by Jaime Sevilla, Pablo Villalobos, Matthew Burtell and Juan Felipe Cerón. We collaborated to add more compute estimates to the public database. I can recommend their first analysis: "Parameter counts in Machine Learning". ↩

9. Transformative AI, as defined by Open Philanthropy in this blogpost: "*Roughly and conceptually, transformative AI is AI that precipitates a transition comparable to (or more significant than) the agricultural or industrial revolution.*" ↩

10. For more thoughts and a discussion on this, I can recommend "The Scaling Hypothesis" by Gwern (or the summary in the AI Alignment Newsletter #156). ↩

11. I would also describe the purple part as an open research question. How can we decompose this — differentiating between parallelization, an engineering effort, and spending, where it is easier to find upper limits? ↩

12. I would be interested in an update on this. However, I also did not spend time looking for an update on this in the recent AI experts surveys. ↩

13. I initially made the claim that there are reasons to believe that the available memory capacity of compute systems might match the human brain or at least be sufficient (at least the information we can consciously recall and access). However, while thinking more about this claim, I became uncertain. I started wondering if the brain also has something similar to a memory hierarchy as it is the default for compute systems (different levels of memory capacities which can be accessed at different speeds). I would be interested in research on this. ↩

14. In general, computational power is key to our modern society, and might also be the foundation of life in the future: digital minds. The future of humanity could be computed on digital computers — see "Digital People Would Be An Even Bigger Deal" by Holden Karnofsky or "Sharing the World with Digital Minds" by Bostrom. ↩

Forecasting Compute - Transformative AI and Compute [2/4]

Cross-posted here on the EA Forum.

Transformative AI and Compute - A holistic approach - Part 2 out of 4

This is part two of the series *Transformative AI and Compute - A holistic approach*. You can find the sequence here and the summary here.

This work was conducted as part of Stanford's Existential Risks Initiative (SERI) at the Center for International Security and Cooperation, Stanford University. Mentored by Ashwin Acharya (Center for Security and Emerging Technology (CSET)) and Michael Andregg (Fathom Radiant).

This post attempts to:

4. Discuss the compute component in forecasting efforts on transformative AI timelines (Section 4)
5. Propose ideas for better compute forecasts (Section 5).

Epistemic Status

This article is *Exploratory* to *My Best Guess*. I've spent roughly 300 hours researching this piece and writing it up. I am not claiming completeness for any enumerations. Most lists are the result of things I learned *on the way* and then tried to categorize.

I have a background in Electrical Engineering with an emphasis on Computer Engineering and have done research in the field of ML optimizations for resource-constrained devices — working on the intersection of ML deployments and hardware optimization. I am more confident in my view on hardware engineering than in the macro interpretation of those trends for AI progress and timelines.

This piece was a research trial to test my prioritization, interest and fit for this topic. Instead of focusing on a single narrow question, this paper and research trial turned out to be *more broad* — therefore *a holistic approach*. In the future, I'm planning to work more focused on a narrow relevant research questions within this domain. Please reach out.

Views and mistakes are solely my own.

Previous Post: What is Compute?

You can find the previous post "*What is Compute? [1/4]*" here.

4. Forecasting Compute

Highlights

- For transformative AI timeline models with compute milestones, we are interested in how much effective compute we have available at year *Y*.
  - We can break this down into (1) **compute costs**, (2) **compute spending**, and (3) **algorithmic progress**.
- **Hardware progress**: For forecasting hardware progress, no single model can explain the improvements of the last years. Instead, a mix of Moore's Law, chip architectures, and hardware paradigms are applicable models and categories to think about progress.
  - Performance improvements can happen significantly faster than the pure improvement in transistor count and density (Moore's law) would indicate.
  - We will see a fragmentation of applications into the *slow* and *fast lane.* High-demand applications will move to the *fast lane* by designing and benefitting from specialized processors. In contrast, low-demand applications will be stuck in the *slow lane* running on general-purpose processors. We should assume that AI will be on the *fast lane*.
- **Economy of scale**: There will *either* be room for improvement in chip design, or chip design will *stabilize* which enables an economy of scale. Our hardware will *first get better* and then *get cheaper.*
- **Hardware spending**: The current increase in spending is not sustainable; however, it could still significantly increase with estimates up to 1% of (US) GDP (a megaproject, like the Manhattan Project or the Apollo program).
  - However, it is still unclear what percentage of the compute trend has been due to increased spending versus performant hardware.

We have discussed that compute is a critical component of AI systems capabilities. Additionally, I have discussed some of the unique properties of compute (compared to data and algorithmic innovation), which make it potentially more *measurable* than the other contributors.

This section will explore the role of compute in an existing Transformative AI timeline model by Cotra (Cotra 2020) and discuss computation hardware, valuable concepts, and the limits of hardware spending.

4.1 Cotra's Transformative AI Timeline Model

I have explored the draft report on AI timelines from Ajeya Cotra and tried to understand the role of compute within this model (Cotra 2020). The following visualization is my try to break down the timeline model and dissect the parts relevant for compute:

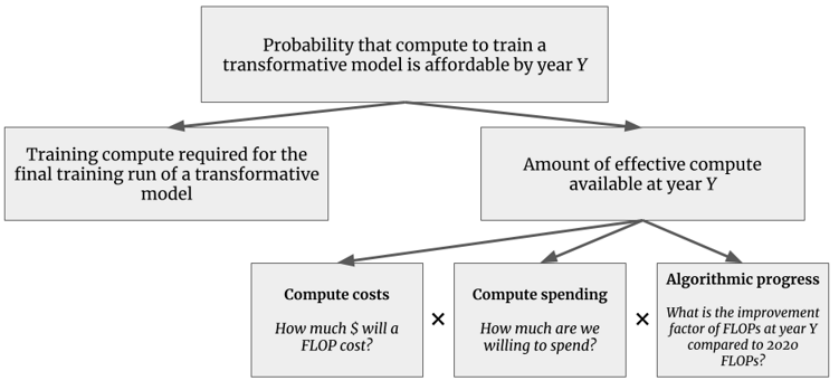


Figure 4.1: The components of Cotra's TAI timeline model.

Overall, we are interested in the probability that the compute to train a transformative model is affordable by year Y. This is informed by:

1. How much compute do we require for the final training run of such a transformative model?
2. The amount of compute which is available/affordable at year Y.

How much compute do we require for the final training run of such a transformative model? This is informed by multiple hypotheses from biological anchors and the compute milestones (see Part 1 - Section 3.5 or the report itself).

The amount of compute which is available/affordable at year Y. This second component is the focus of this report. It is divided into three components:

1. **Compute costs**: How much $ will a FLOP cost?
2. **Compute spending**: How much are we willing to spend?
3. **Algorithmic progress**: What is the improvement factor of FLOPs at year Y compared to 2020 FLOPs?

(1) Compute cost: What is the price for a FLOP? Assuming we have a specific budget from (2), how many FLOPs can we buy and spend on our final training run? Cotra assumes a hardware utilization[1] of ≈⅓. (Here is the section in the draft report.)

(2) Compute spending: How much are governments and/or organizations willing to spend on an AI systems' final training run? (Here is the section in the draft report.)

(3) Algorithmic progress: As discussed in Part 1 - Section 3.3, our AI systems get *more capabilities* per FLOP over time — that is algorithmic progress or efficiency. Consequently, as we estimate compute and not effective compute, we need to adjust our compute estimate by a relevant factor over time. You can think of it like: How much better is a FLOP in year Y than a 2020 FLOP? (Here is the section in the draft report.)

Those three components are then multiplicative:

$$\text{Effective Compute}_{\text{Year Y}}[\text{FLOP 2020}] = \text{Compute Costs}[\frac{\text{FLOP}_{\text{2020}}}{\text{2020\$}}] \times \text{Compute Spending}[\text{2020 \$}] \times \text{Efficiency Gains}[\frac{\text{FLOP}_{\text{2020}}}{\text{FLOP}_{\text{Year Y}}}]$$

All three components are modeled as logistic curves — assuming that they are improving at some exponential constant rate but will saturate in the future (Cotra 2020).

Overall, I can strongly recommend reading through the report, as it is the most detailed work on AI timelines to my knowledge available. There is a summary for the AI Alignment newsletter and you can find the model in this spreadsheet.

Instead of discussing the whole piece, the following subsections will discuss some concepts on how to think about (1) compute prices and (2) compute spending. I will also present Cotra's forecasts.

4.2 Forecasting Computing Prices

Computing prices are informed by the purchase of computing hardware, energy costs, and potentially connected engineering time. As a useful proxy one can think about renting computational power from cloud services, such as Google Cloud or Amazon Web Services (AWS). The compute performance times the hourly renting rate brings access to a quantity of compute for a set price. This price would include all the costs, including a markup to be profitable.

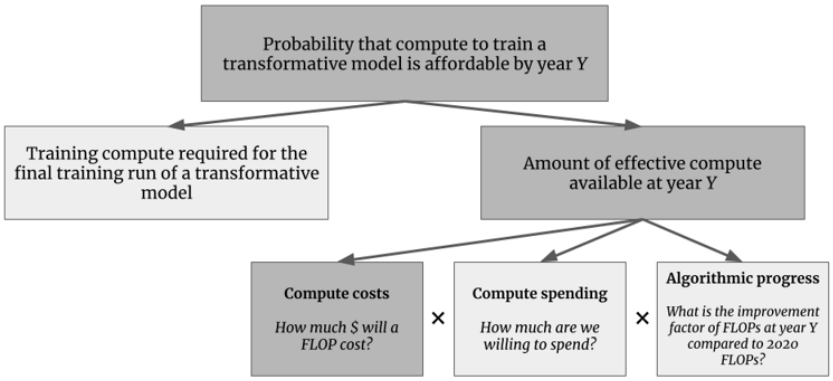


Figure 4.2: Hardware prices as a component for forecasting effective compute.

As a proxy metric FLOP/$ is often used. We have two options to *advance* in this domain:

- **Better**: Our computing hardware achieves more FLOP/S (for the same price).
- **Cheaper**: Our computing hardware gets cheaper (while having the same amount of FLOP/S).

For thinking about those trends, I will be discussing Moore's Law, chip architectures, and hardware paradigms. Those trends did and might lead to more FLOP per $.

Moore's Law

A common way to model progress in computing hardware is Moore's law. It is probably the most well-known and also commonly used outside the research domain. It is mainly used to describe the exponential growth of technology — sometimes more precisely in the manner of *every two years the computing power doubles.*

Both are wrong but capture an interpretation that is becoming *less accurate*. The original Moore's law quotes:

> "Moore's law is the observation that the number of transistors in a dense integrated circuit (IC) doubles about every two years."

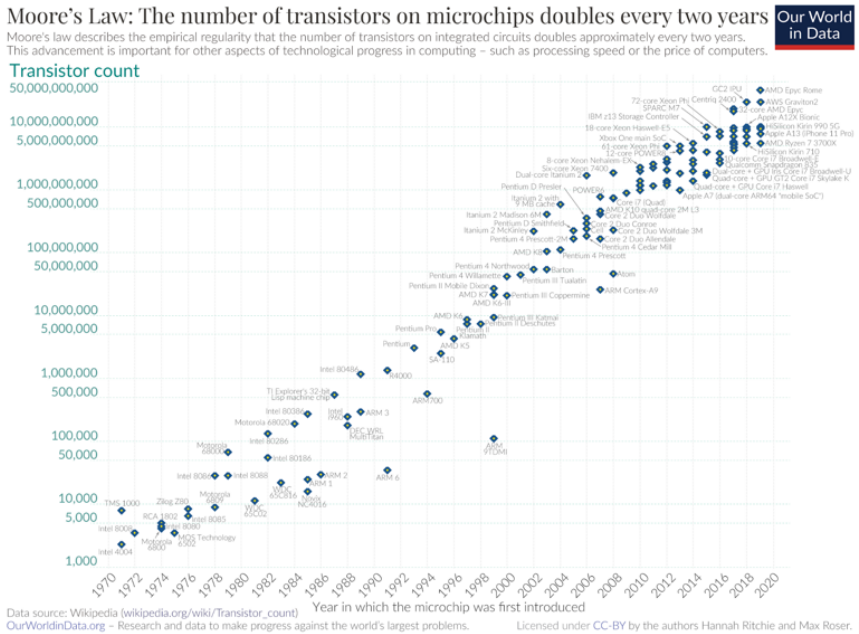


Figure 4.3: (Original) Moore's Law — the number of transistors on computing hardware over time. (Taken from Our World in Data.)

Transistors are the fundamental building block of integrated circuits (IC) (or chips), so having more transistors in an IC is advantageous. However, in the end, for our forecast, we care about increased performance or reduced costs — and Moore's Law does not describe this directly. It is a direct driver for efficiency (power use of each transistor) but not for performance.

Having more building blocks available can build more memory, more processing cores, and others, and those can lead to speed improvements but not necessarily *need to*.

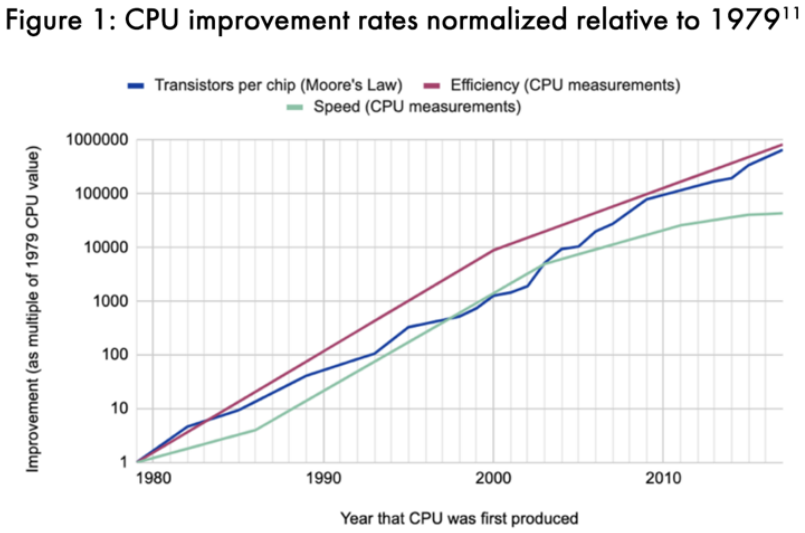


Figure 4.4: CPU improvements rates[2] normalized relative to 1979. (Taken from (Khan 2020).)

Figure 4.4 shows the normalized improvements of transistors per chip (blue), the efficiency (red), and the speed (green). The central insight is that the doubling rate of Moore's law did a pretty good job describing the efficiency and speed improvements until 2005, but CPU speed could not maintain this trend.

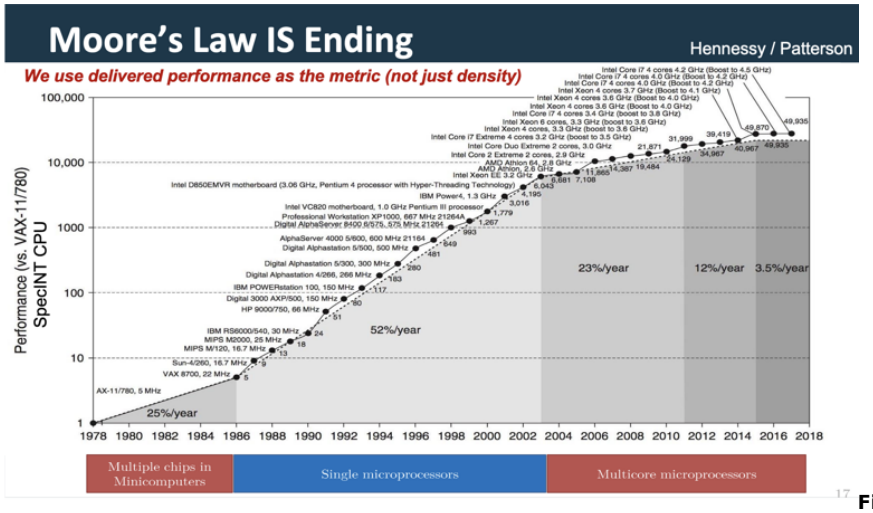


Figure 4.5: The performance of CPUs (in SPECint, a standard CPU benchmark) over time depicting the different growth rates per year over the three eras: multiple chips in minicomputers, single microprocessors, and multi-core microprocessors. (Taken from (Shalf 2020b).)

The performance trend of CPUs could not be maintained over time and we see a decreasing yearly growth rate. This is partially explained by the end of eras as the previous trend could not be scaled (single microprocessor) and new directions were more complex and not a pure hardware job, such as multicore-architectures (Figure 4.5).

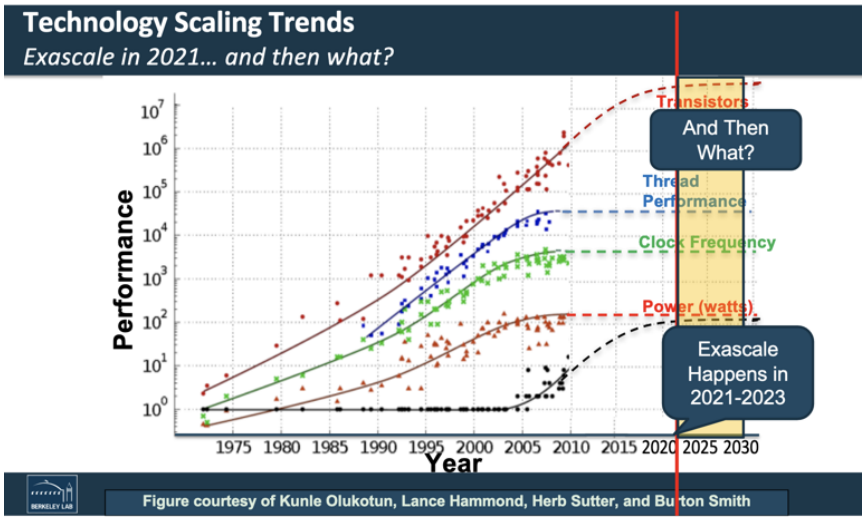


Figure 4.6: The performance over time of transistors, threads, clock frequency, power and number of cores. (The black dots and trend line depict the number of cores.) (Taken from (Shalf 2020b).)

Dissecting this trend into other components reveals some of the progress drivers and the start of new eras (Figure 4.6).

Moore's Law Conclusion

Long story short, *Moore's law end's* — you have heard this before. This is not my message here. Moore's law was a useful proxy for the overall performance and progress of technology, and it still sometimes is (independent of its literal meaning). Despite that, Moore's Law does not capture trends across computing paradigms or predominator architecture types (as seen in Figure 4.5 and 4.6) — it is useful *within an era*. For Moore's law this era was the single microprocessor one.

In the previous section, we have discussed an exponential growth in the compute used for AI systems. However, we have just seen that actual performance growth is decreasing. Explaining the remaining AI compute growth with increased spending would be mistaken. Moore's law does not capture chip architectures — such as the switch from CPUs to GPUs with AlexNet; and this is a predominant factor for growth in computing power, and consequently, compute. Therefore performance improvements can happen significantly faster than the pure improvement in transistor count and density would indicate.

To understand progress in computing hardware, especially within AI systems, the model of Moore's law provides minimal information value. For describing a doubling of computing performance, we can simply tell it as a doubling of computing performance (or you can come up with your own law[3]). Dissecting any performance growth will always be complex (remember our three basic components: logic, memory and interconnect) but if we do so, we should focus on the most important contributors on higher abstraction layers[4].

Chip Architectures: From Flexibility to Efficiency

In Part 1 - Section 2.2, we have discussed that a new era, the modern era, was introduced by AlexNet leveraging a certain chip architecture[5]: GPUs. Graphical processing units (GPUs) are a type of integrated circuit which were originally designed for computing graphics to then be displayed. Typical operations were rendering polygons and other geometric calculations — operations which are mathematically mapped as matrix multiplications. The partial results of the matrix multiplications can be computed independent from one another which allows a parallelized computation, and hence, the original design of a GPU: a chip architecture with highly parallelized computing units. Due to the highly parallel architecture of GPUs they are also well suited for non-graphical computations: for *embarrassingly parallel* problems, such as training neural nets.

Figure 4.7: The spectrum of chip architectures with trade-offs in regards to efficiency and flexibility. (Taken from (Microsoft Documentation 2020).)

GPUs are one example of such a chip architecture across a spectrum. While GPUs are somewhat specialized due to their parallelized architecture, they can still execute various workloads. This makes them more efficient than our general-purpose processors, CPUs, and less flexible regarding their workload (Figure 4.7).

Walking further on the spectrum towards less flexible (more specialized) but more efficient, we find architectures such as field-programmable gate arrays (FPGAs) and application-specific integrated circuits (ASICs). As the name of an ASICs implies, it is a

dedicated hardware architecture for a specific application — making it specialized (and not flexible at all) but highly efficient at the dedicated task.

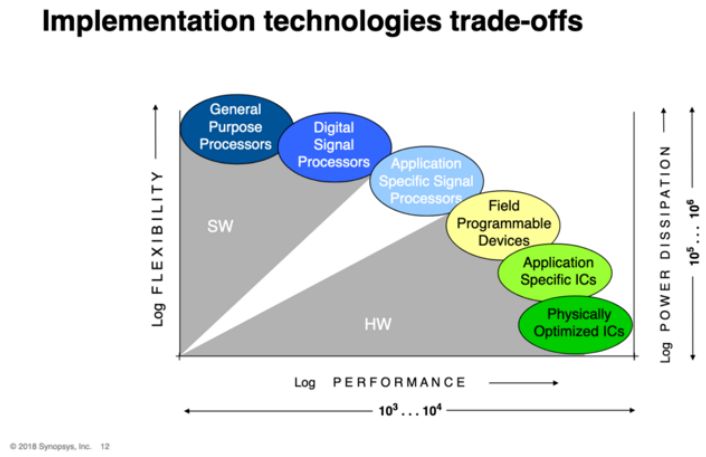


Figure 4.8: Complexity-flexibility trade-offs for different approaches for implementation in digital systems. (Taken from lecture slides of Electronic Design Automation class at RWTH Aachen.)

Thompson et al. discuss in "The decline of computers as a general purpose technology" the trend from general-purpose computing towards specialization. They differentiate modern systems between the *fast lane* and *slow lane.* While we can execute all kinds of applications and tasks on general-purpose processors, we can tailor ASICs to a specialized workload. For applications where it is economically feasible to create specialized processors, they are on the *fast lane*. There is enough demand, and the application's workload is specific enough to benefit from dedicated hardware.
Other applications with diverse workloads and less demand have to rely on general-purpose processors and do not benefit from the efficient and accelerated ICs — this is the *slow lane* (Thompson and Spanuth 2021).

An example of an application being on the fast lane is Bitcoin mining. Bitcoin mining has relied for years now on ASICs due to the highly specific nature of the mining operation. It is economically not sustainable to mine without dedicated hardware, as those mining ASICs are orders of magnitude (OOM) more efficient.

Chip Architectures Conclusion

Moving from general-purpose processors to highly specialized ones if the economic incentives allow is a recent trend and useful model for classifying and thinking about progress. We have seen for applications that are *specific* enough and demand, such as Bitcoin mining, the industry moves towards specialized processors, and competing without is not feasible.

For AI systems, we have seen that moving from CPUs towards GPUs has enabled the field and resulted in a new compute growth trend. However, we have not seen "the ASICs" of AI systems yet. Clearly, the economic incentives align, as the AI market is big enough, and the trend is similar old as Bitcoin. One problem is the clearly defined application or workload. Whereas bitcoin mining simply executes the same block of operations that make up the function SHA-1, for AI systems, this workload is not clearly defined — there is not *the one AI function*. However, we might see a convergence in the future towards certain basic building blocks and operations, where *more* specialized processors might be feasible. I will not discuss the details of this trend and the feasibility but would be interested in an investigation (see Appendix A). This is an important concept to monitor. I am not saying we will move further on the spectrum for the whole domain — we might not, as AI is too general. However, we might go for a specific AI system's architecture, and if we do so, we can expect OOM of progress (Figure 4.9).

Table 2: Comparing state-of-the-art AI chips to state-of-the-art CPUs

	Training		Inference		Generality[88]	Inference accuracy[89]
	Efficiency	Speed	Efficiency	Speed		
CPU	1x baseline				Very High	~98-99.7%
GPU	~10-100x	~10-1,000x	~1-10x	~1-100x	High	~98-99.7%
FPGA	-	-	~10-100x	~10-100x	Medium	~95-99%
ASIC	~100-1,000x	~10-1,000x	~100-1,000x	~10-1,000x	Low	~90-98%

Table 4.9: Comparing state-of-the-Art AI chips to state-of-the-art CPUs. (Taken from (Khan 2020).)

We have seen trends moving further from the GPU towards the ASIC on the spectrum, e.g., Google's TPU.

Hardware Paradigms

Our dominant hardware paradigm is and was integrated circuits for the last 50 years — given that it is currently our most efficient one.

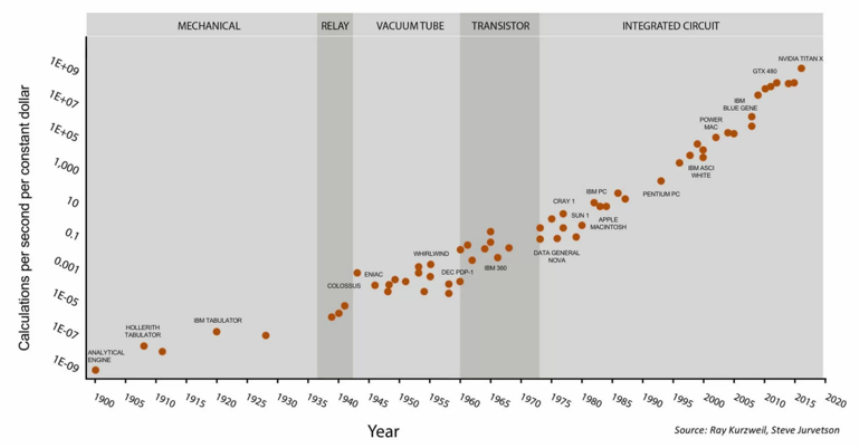


Figure 4.10: The performance over time is categorized in different hardware paradigms, such as mechanical, relay, vacuum tube, transistors, and most recently, integrated circuits. (Taken from (Cotra 2020) and Wikimedia.)

Nonetheless, we explore alternative paradigms (in parallel) over time, and once they are more efficient (and cheaper) than the current dominant paradigm, the field as a whole will transition towards that paradigm, such as optical computing, quantum computing, or others. Consequently, one should continuously evaluate the new hardware paradigms on the horizon, estimate their computational gains, and feasibility within the next decade.

Hardware in Cotra's TAI Timeline Model

Cotra assumes a doubling time of 2.5 years and then leveling off after 6 OOM of progress for FLOP per $ by 2100. This is half as many OOM of progress we had over the last 60 years (1960s to 2020s).

Cotra's goes into more detail in the appendix. She acknowledges that this forecast is probably the least informed piece of her model[6]. Nonetheless, there are various ideas which we can relate to our previous sections.

Here is the brief summary of how a ≈144-fold increase could be achieved in the next ≈20 years (the factors are multiplicative). We can get better by:

- Increasing transistors efficiency (traditional Moore's Law)
  - Factor of ≈3
- Deep learning specific chip design choices (such as low precision computing, addressing the communication bottleneck (interconnect))
  - Factor of ≈6

And, we can get cheaper by the stabilization of chip design and enabling an economy of scale:

- Semiconductor industry amortize their R&D cost due to slower improvements
  - Factor of ≈2
- Sale price amortization when improvements are slower
  - Factor of ≈2
- A combination of economies of scale, greater specialization and more intense optimization (reducing the amortize cost of GPUs as compared to the power consumption cost)
  - Factor of ≈2

Overall, there will *either* be room for improvement in chip design, or chip design will *stabilize* which enables the above outlined improvements in the economy of scale (learning curves). Consequently, if you believe that technological progress (more performance for the same price) might halt, the compute costs will continue decreasing, as we then get *cheaper* (same performance for a decreased price).

4.3 Forecasting Compute Spending

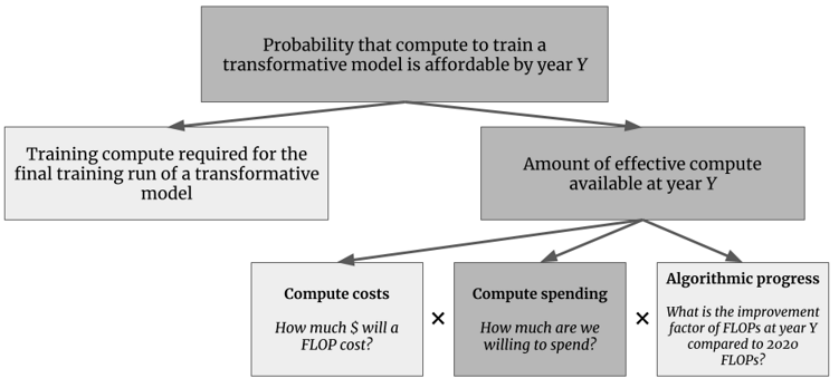


Figure 4.11: Compute spending as a component for the amount of effective compute available.

Once we know the price of compute, we need to estimate the potential spending. We will quickly discuss the estimate of Cotra's model and the importance of dissecting progress into hardware improvements, increased spendings, and their economic limitations.

Compute Spending in Cotra's TAI Timeline Model

For hardware spending, we estimate the maximum amount (2020 $) an actor is willing to spend on the final training run. The current estimate for the most expensive run in a published paper was the final training run for AlphaStar at roughly $1M.

Cotra compares the maximum a single actor is willing to spend to a mega-project, such as the Manhattan Project or the Apollo Program, around 0.5% of the US GDP for four years. As AI will be more economically and strategically valuable, she estimates the spending with up to 1% of GDP of the largest country for up to 5 years (assuming GDP is growing at ≈3% per year).

Economical limits

We have discussed before (in Section 2.2) that it is unclear how much of the increase in compute is due to more performant hardware or increased spending. If we assume that spending is capped at a certain percentage (e.g., at 1% of US GDP), those trends are not sustainable if increased spending is the dominant component.

Carey discusses this in "Interpreting AI compute trends" and breaks down the trend by dissecting it in FLOPS per $ and increased spending.[7] He extrapolates the trend of spending and estimates an end of the trend in 3.5 to 10 years (from 2018) — also assuming a Megaproject with 1% of US GDP spending (Carey 2018).

Thompson et al. also discuss this and conclude that the current trend is "economically, technically and environmentally unsustainable", as the requirement for computation for the training is outperforming hardware performance (Figure 4.12) (Thompson et al. 2020).

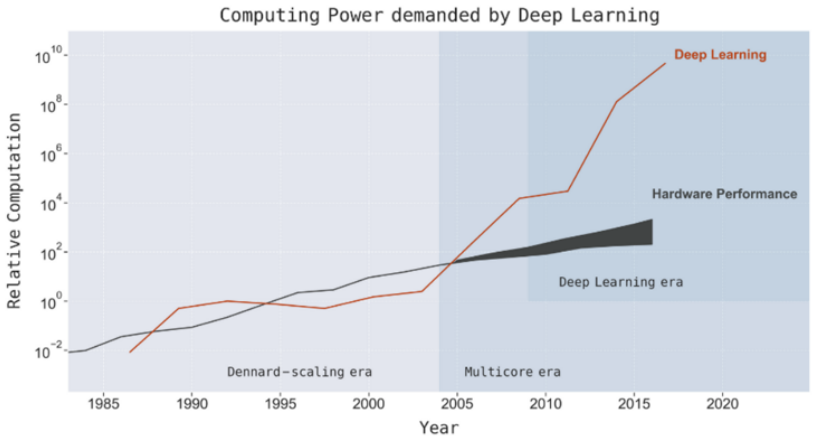


Figure 4.12: Computing power demanded by deep learning. (Taken from (Thompson et al. 2020).)

4.4 Forecasting Algorithmic Progress

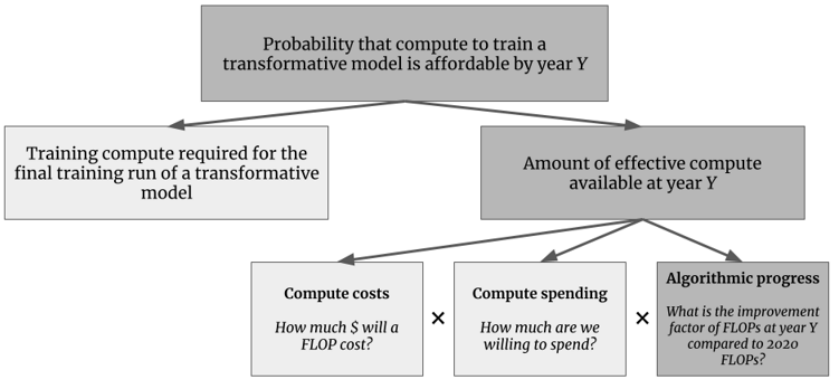


Figure 4.13: Algorithmic progress as a component for the amount of effective compute available.

Forecasting algorithmic progress is out of scope for this piece. However, for the sake of completeness, I list it here. It is significant for the *effective* amount of compute available. As previously discussed, the best piece available is "AI and Efficiency". For ImageNet, the algorithmic improvements halve the compute required for the same task every 16 months (Hernandez and Brown 2020).

More research in algorithmic efficiency is crucial for estimating effective compute available and the current public research in this domain is limited.

Algorithmic Progress in Cotra's TAI Timeline Model

Cotra estimates the algorithmic progress with a halving time of 2 to 3 years with a maximum of 1 to 5 OOM.

4.5 Conclusion

We have discussed the role of compute for Transformative AI timelines and dissected the relevant components. For improving forecasting efforts breaking it down into conceptual blocks is useful. We have focused on three concepts for compute prices: Moore's law, chip architectures, and hardware paradigms, which provide valuable insights for thinking about progress and, consequently, forecasting it. This piece does not outline a concrete strategy for forecasting *compute*. Rather, it lists various ideas and discusses them related to Cotra's TAI timeline.

Considering the economy of scale is an important model which might allow cheaper hardware even though our technical and hardware progress might be reduced to a smaller extent and longer innovation cycles. Additionally, the switch in hardware paradigms is essential to monitor. The Appendix A lists some research questions related to this.

Overall, I think the compute concept of Cotra's TAI timeline model is impressive work and an excellent method to estimate effective compute available at year *Y* (I cannot comment on the other parts of the model). Improving the hardware estimates is a possible and potential part of my future work. In the next section, I will outline some concrete proposals which might allow us to come up with better compute price estimates.

5 Better Compute Forecasts

Highlights

- Researchers should share insights into their AI system's training by disclosing the amount of compute used and its connected details.
    1. Ideally, we should make it a requirement for publications and reduce the technical burden of recording the used compute.
- For forecasting hardware progress, we can rely on conceptual models and categories of innovation. We can break this down into:
    1. Progress in current computing paradigms
    2. Economy of scale for existing technologies
    3. Introduction of new computing paradigms
    4. Unknown unknowns
- We should also monitor the dominant design strategy of hardware as this informs our forecasts. The three design strategies are (1) hardware-driven algorithm design, (2) algorithm-driven hardware design, and (3) co-develop hardware and algorithm.
- Metrics, such as FLOPS/$, often give limited insights, as they only represent one of the three computer components (memory, interconnect, and logic). Understanding their limitations is essential for forecasting.

We have discussed how to forecast future compute progress. However, these forecasting efforts could easily be more informed. In this section, I will sketch some proposals which could improve forecasting efforts.

As discussed, Ajeya Cotra also thinks that the compute part could be easily improved and she “would be very excited” for people to take up the open questions around hardware progress and more in-depth analyses, and I agree.[8] Also Holden Karnofsky would be interested in better compute forecasts.[9]

This section presents some of my and others' thoughts on the topic, but they could easily be improved with better data, as well as efforts to improve our collective mental models and technical understanding of AI hardware. In this section, I offer some specific suggestions for how to improve our understanding of this area. The listed proposals are in no particular order.

5.1 Share AI System Training Insights

I would like to ask AI researchers to publish data on their compute usage. Ideally, researchers should publish the following metrics, for the final training run, and, preferably, also for the development stages:

- Amount of compute used in FLOPs/OPs.
- The number representation used (float16, float32, bfloat16, int8, etc.)
- The computing system and hardware used: type of GPUs, number of GPUs, and the networking of the system.
    - Ideally a metric for the utilization of the system.
- The software and optimization stack (compilers etc) used.
- The rough amount of money spent on the compute.

This is, of course, easier said than done. I do think that most researchers are not aware of the information value of those data points for the field of macro ML research.[10] Also, accessing data, such as compute used, is not easily accessible and often *hidden behind layers of complexity* for ML researchers. Therefore, it would be highly valuable to develop a plugin that allows researchers to access the used compute, the same way as they access validation loss or other key metrics. One could also imagine making the publication of those metrics required for the publication at top conferences — similar to the societal impact statement for NeurIPS (Centre for the Governance of AI 2020).

5.2 Components of Hardware Progress

When forecasting the compute prices or the price per FLOP, it is desirable to understand the categories *where progress might happen*. We can break this down into:

- Additional progress in current computing paradigms
- Economy of scale in the current dominant computing paradigm
- Introduction of new computing paradigms

And, of course, we always have *unknown unknowns*.

Additional progress in current computing paradigms describes the continued improvement in the digital computing domain using transistors on semiconductors. This itself is a broad category and the category where we have seen the progress of the last 50 years. Examples are architectural innovations (from CPU to ASICs), smaller transistors, post-CMOS, other semiconductor material innovations, high-performance computation communication, and many more.

Economy of scale in the current dominant computing paradigm describes the reduced costs when innovation periods are extended and amortization happens (as discussed in Section 4.2). The recent innovation cycles are relatively short, which leads to higher prices. Once innovation stalls, we could enter a period of amortization which enables an economy of scale which drives price further down. At this point, the energy costs could also become the dominant cost factor of compute (Cotra 2020).

The introduction of new computing paradigms characterizes a new upcoming computing paradigm that is not yet economically feasible but might in the future be more efficient than the currently dominant one. Examples are quantum computing, optical computing, neuromorphic computing, and others.

And just to be safe, **unknown unknowns**.

Design Strategies

Those innovations within hardware progress components are then enabled by different design strategies. The design strategy describes *how* we innovate. I think this is also a helpful model to classify progress and model for hardware innovation. Shalf outlines three different design strategies (Shalf 2020a):

- Hardware-drive algorithm design
- Algorithm-driven hardware design
- Co-develop hardware and algorithm

Hardware-driven algorithm design describes the modification of algorithms to take advantage of existing and new accelerators. Examples are the usage of GPUs where we adapt our training algorithms for deployment on GPUs.

Algorithm-driven hardware design is hardware specifically designed for specific workloads with dedicated accelerators based on the application and its algorithms. However, the development costs are way higher. As discussed before, this is the *fast lane* and only available when the market demand (and application) allows it.

Co-develop hardware and algorithms describes more of an economic model of interacting with the semiconductor industry. The previous general-purpose computing has led to a handoff relationship. However, various innovations have led to multiple companies co-designing their hardware with software within the last few years. Examples in the AI systems are various chips for smartphones and Google TPUs, or Tesla’s recent AI chip announcement: Dojo. This design strategy seems to have become a trend for actors which can afford so.

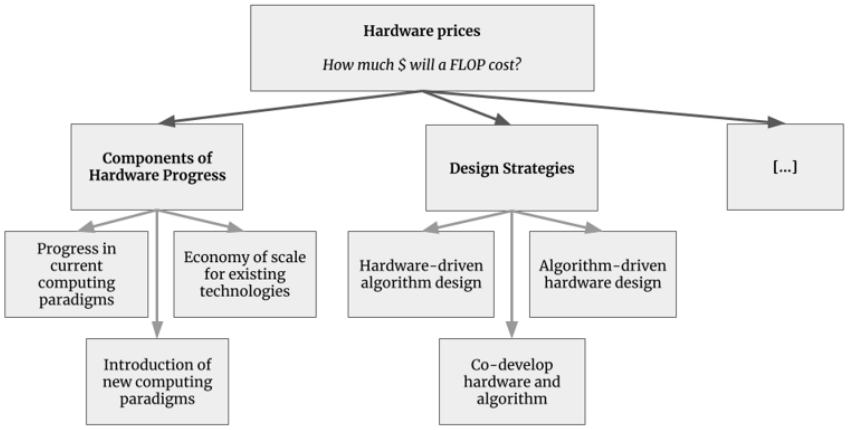


Figure 5.1: Components and models for conceptualizing hardware progress.

5.3 Conclusion

All in all, forecasting compute is a major undertaking, and this piece does not outline a concrete strategy on how to go about it; instead, it lists various ideas. I see it as a start for potentially building a bigger conceptual model and providing estimates and forecasts on the subcategories. I would recommend doing the same for hardware spending.

For a discussion on common metrics, see Appendix B.

Next Post: Compute Governance and Conclusions

The next post "*Compute Governance and Conclusions [3/4]*" will attempt to:

6. Briefly outline the relevance of compute for AI Governance (Part 3 - Section 6).
7. Conclude this report and discuss next steps (Part 3 - Section 7).

Acknowledgments

You can find the acknowledgments in the summary.

References

The references are listed in the summary.

---

1. We have discussed under and overutilization in Part 1 - Section 1. Compute hardware is usually underutilized due to the processor-memory performance gap. ↩
2. For transistor data, see Max Roser and Hannah Ritchie, “Technological Progress,” Our World in Data, 2019, https://ourworldindata.org/technological-progress. For efficiency data, see Koomey et al., “Energy Efficiency of Computing." For speed data, see Hennessy et al., “New Golden Age,” 54. ↩
3. For example there's Huang's Law: “*Huang's Law is an observation in computer science and engineering that advancements in GPUs are growing at a rate much faster than with traditional CPUs.*” ↩
4. With higher abstraction layers I refer to the abstraction layers of computing systems. From electrons in a semiconductor to software. Higher abstraction layers are usually easier to translate into performance growth, whereas an innovation in transistors is hard to *translate over all the layers into a performance increase.* ↩
5. The word *architecture* is used for describing various hardware concepts on different layers. In this piece with chip architecture, I refer to the spectrum from CPU to ASICs — while all of them still rely on digital computing and semiconductor material. ↩
6. “*Because they have not been the primary focus of my research, I consider these estimates unusually unstable, and expect that talking to a hardware expert could easily change my mind.*” ↩
7. By using the dataset “2017 trend in the cost of computing” by AI Impacts. While I agree with the overall premise that spending will be capped and has been a major driver, I am skeptical of using the AI Impacts dataset for estimating cost of computing trends. I will be discussing some caveats on commonly used metrics in Appendix B. ↩
8. AXRP Episode 7.5 (released in May 2021; only the transcript is available):

"[...] but I would be very excited for a piece that was just like, here is where hardware progress will tap out and here's when I think that will happen and why."

"I think an easier in-road is trying to answer one of the many open questions in the timelines report. Trying to really nail hardware forecasting, or really nail algorithmic progress some way. And I think that if it's good is adding direct value and it's also getting you noticed."

↩

9. 80'000hours episode #109 (released in August 2021):

   "A thing that would really change my mind a lot is if we did the better compute projections — I'm expecting that to come out to similar conclusions to what we have at the moment; different, but similar — but maybe we did them and it was just like [...]"

   ↩

10. While I agree that our estimates (as discussed in Part 1 - Section 2.3) might be *good enough* for trends over multiple years (assuming that our error is within one OOM), having access to the exact numbers is beneficial because:

    1. It makes the data acquisition process easier and faster.
    2. The current estimates use different methods depending on the published data (based on GPU days and assuming utilization, based on the model size/number of parameters, or based on the inference compute cost).
    3. There is more than "just the final training run". Other processes of the development might play a role (or at least should be explored): fine-tuning the hyper-parameters, neural architecture search, and others.
    4. We do not have any insights in the breakup into increased spending or better hardware.

    ↩

Compute Governance and Conclusions - Transformative AI and Compute [3/4]

Cross-posted here on the EA Forum.

Transformative AI and Compute - A holistic approach - Part 3 out of 4

This is part two of the series *Transformative AI and Compute - A holistic approach*. You can find the sequence here and the summary here.

This work was conducted as part of Stanford's Existential Risks Initiative (SERI) at the Center for International Security and Cooperation, Stanford University. Mentored by Ashwin Acharya (Center for Security and Emerging Technology (CSET)) and Michael Andregg (Fathom Radiant).

This post attempts to:

6. Briefly outline the relevance of compute for AI Governance (Section 6).
7. Conclude this report and discuss next steps (Section 7).

Epistemic Status

This article is *Exploratory* to *My Best Guess*. I've spent roughly 300 hours researching this piece and writing it up. I am not claiming completeness for any enumerations. Most lists are the result of things I learned *on the way* and then tried to categorize.

I have a background in Electrical Engineering with an emphasis on Computer Engineering and have done research in the field of ML optimizations for resource-constrained devices — working on the intersection of ML deployments and hardware optimization. I am more confident in my view on hardware engineering than in the macro interpretation of those trends for AI progress and timelines.

This piece was a research trial to test my prioritization, interest and fit for this topic. Instead of focusing on a single narrow question, this paper and research trial turned out to be *more broad* — therefore *a holistic approach*. In the future, I'm planning to work more focused on a narrow relevant research questions within this domain. Please reach out.

Views and mistakes are solely my own.

Previous Post: Forecasting Compute

You can find the previous post "Forecasting Compute [2/4]" here.

6. Compute Governance

Highlights

- Compute is a unique AI governance node due to the required physical space, energy demand, and the concentrated supply chain. Those features make it a governable candidate.
- Controlling and governing access to compute can be harnessed to achieve better AI safety outcomes, for instance restricting compute access to non-safety-aligned actors.
- As compute becomes a dominant factor of costs at the frontier of AI research, it may start to resemble high-energy physics research, where a significant amount of the budget is spent on infrastructure (unlike previous trends of CS research where the equipment costs have been fairly low).

Lastly, I want to motivate the topic of compute governance as a subfield of AI governance and briefly highlight the unique aspect of compute governance.

Compute has three unique features which might make it *more governable* than other domains of AI governance (such as talent, ideas, and data) (Anderljung and Carlier 2021):

1. Compute requires physical space for the computing hardware — football-field-sized supercomputer centers are the norm (Los Alamos National Laboratory 2013). Compared to software, this makes compute easier to track.
    - Additionally, compute is often highly centralized due to the dominance of cloud providers, such as Amazon Web Services (AWS), Google Cloud, and others. Moreover, current leading hardware, such as Google TPUs, is only available as a service. Consequently, this feature makes it more governable.
2. The energy (and water demands). For running those supercomputers, massive amounts of energy and water for cooling are required (Los Alamos National Laboratory 2013).
3. The supply chain of the semiconductor is highly concentrated, which could enable monitoring and governance (Khan 2021) — see "The Semiconductor Supply Chain" by CSET for more.

Second, according to my initial research and talking to people in the field of AI governance, there seems to be more of a consensus on *what to do with compute* regarding governance: restricting and regulating access to compute resources for less cautious actors.[1] This does not include a consensus on the concrete policies but at least in regards to the goal. Whereas for other aspects in the field of AI governance, there seems to be no clear consensus on which intermediate goals to pursue (see a discussion in this post).

6.1 Funding Allocation

Within this decade, we will and should see a switch in funding distribution at publicly funded AI research groups. Whereas AI and computer science (CS) research groups usually had relatively low overhead costs for equipment, this will change in the future to the increased need for spending more funding on compute to maintain state-of-the-art research. Those groups will become more like high-energy physics or biology research groups where considerable funding is being spent on infrastructure (e.g., equipment and hardware). If this does not happen, publicly funded groups will not be able to compete. We can already observe this *compute divide* (Ahmed and Wahed 2020).

6.2 Research Questions

For a list of research questions see some “Some AI Governance Research Ideas” (Anderljung and Carlier 2021). My research questions are listed in Appendix A, including some notes on compute governance-related points.

7. Conclusions

Highlights

- In terms of published papers, the research on compute trends, compute spending, and algorithmic efficiency (the field of macro ML research) is minor and more work on this intersection could quickly improve our understanding.
- The field is currently bottlenecked by available data on macro ML trends: total compute used to train a model is rarely published, nor is spending. With these it would be easier to estimate algorithmic efficiency and build better forecasting models.
- The importance of compute also highlights the need for ML engineers working on AI safety to be able to deploy gigantic models.
    - Therefore, more people should consider becoming an AI hardware expert or working as an ML engineer at safety-aligned organizations and enabling their deployment success.
- But also working on the intersection of technology and economics is relevant to inform spending and understanding of macro trends.
- Research results in all of the mentioned fields could then be used to inform compute governance.

Compute is a substantial component of AI systems and has been a driver of their capabilities. Compared to data and algorithmic innovation, it provides a unique quantifiability that enables more efficient analysis and governance.

The effective available compute is mainly informed by the compute prices, the spending, and algorithmic improvements. Nonetheless, we should also explore the downsides of purely focusing on computational power and consider using metrics based on our understanding of the interconnect and memory capacity.

We have discussed components of hardware progress and discussed the recent trends such as Moore’s law, chip architectures, and hardware paradigms. Focusing on only one trend comes with significant shortcomings; instead, I suggest we inform our forecasts by combining such models. I would be especially excited to break down existing compute trends into hardware improvements and increased spending.

Limited research in the field of macro AI

My research is based on a small set of papers, whereas most focus on certain sub aspects. Overall, the research field of macro ML trends in used compute is, to my understanding, fairly small. Seeing more research efforts on compute trends and algorithmic innovation could be highly beneficial. This could lead to a better understanding of past trends, and forecasting future trends — for example, breaking down the trend into increased spending and hardware progress can give us some insights into potential upper limits.

Limited data for analyzing AI trends

Another limitation, and perhaps the cause of limited research, is that , there is also limited data available. Consequently, researchers first need to build the required dataset. I would be excited to see bigger datasets of compute requirements or experiments to measure algorithmic efficiency.

We share in this work our public ML progress dataset and a dataset using MLCommons training benchmarks (MLCommons 2021) for measuring the performance progress of modern AI hardware and ask others to share their insights and data.

ML deployment engineers

As the role of compute is significant for AI progress, there is a strong need for ML engineers who can efficiently deploy AI systems. This was also discussed by Olah in an 80’000 hours episode #107. Consequently, ML engineers should consider working at safety-aligned organizations and enable the deployment of gigantic models which are —ideally— reliable, interpretable and steerable.

Interdisciplinary research

An essential component for compute prices and spending are economic models — either based on spending, or the computing industry, such as the semiconductor industry. Interdisciplinary research on those questions could be of great benefit. Examples of such work are (Thompson et al. 2020; Thompson and Spanuth 2021).

I plan to work on aspects of this research in the future and would be especially interested in exploring collaboration or other synergies. Please reach out. The exact research questions are still to be determined.

Appendix A lists various research questions that I would be interested in exploring and also want others to explore.

Next Post: Compute Research Questions and Metrics

The appendix "*Compute Research Questions and Metrics [4/4]*" will attempt to:

8. Provide a list of connected research questions (Appendix A).
9. Present common compute metrics and discusses their caveats (Appendix B).
10. Provide a list of Startups in the AI Hardware domain (Appendix C).

Acknowledgments

You can find the acknowledgments in the summary.

References

The references are listed in the summary.

1. It seems reasonable and somewhat likely to me that we will be regulating and restricting the export of AI hardware even harsher and might classify it legally as weapons within the next decades. ↩

Compute Research Questions and Metrics - Transformative AI and Compute [4/4]

Cross-posted here on the EA Forum.

Transformative AI and Compute - A holistic approach - Part 4 out of 4

This is the Appendix (part four) of the series *Transformative AI and Compute - A holistic approach*. You can find the sequence here and the summary here. This appendix attempts to:

8. Provide a list of connected research questions (Appendix A).
9. Present common compute metrics and discusses their caveats (Appendix B).
10. Provide a list of Startups in the AI Hardware domain (Appendix C).

Previous Post: Compute Governance and Conclusions

You can find the previous post "Compute Governance and Conclusions [3/4]" here.

Appendix

This appendix lists research questions, some thoughts on metrics related to compute, and a list of AI hardware startups.

These paragraphs ended up in the Appendix, as I have spent less time on them and it is more of a draft. Nonetheless, I still think they can be potentially useful.

A. Research Questions

This is not a research agenda or anything close to it. This is me doing research and thinking "*Oh, I'd be curious to learn more about this *scribbling it down**".

The questions rank from specific to nonspecific, from important to *"I'm just curious".* There are a couple of questions which I think are more important, I've marked them with a ★.

This list is also published as a Google Doc, so one can add comments to individual items.

In general, this domain of research requires more:

- **Data Acquisition**: Collect data in a machine-readable format on compute trends, hardware performances, algorithmic efficiency,

    - Parameter, Compute and Data Trends in Machine Learning (Sevilla et al. 2021) is such an example. Work with us!
- **Data Analysis**: Data analysis is a crucial ingredient: from trend graphs and doubling times up to new measurements of AI systems' capabilities and AI hardware.
- **Data Interpretation**: Once we have available data and analyzed general trends, we require interpretation for informing forecasts and potential policies.

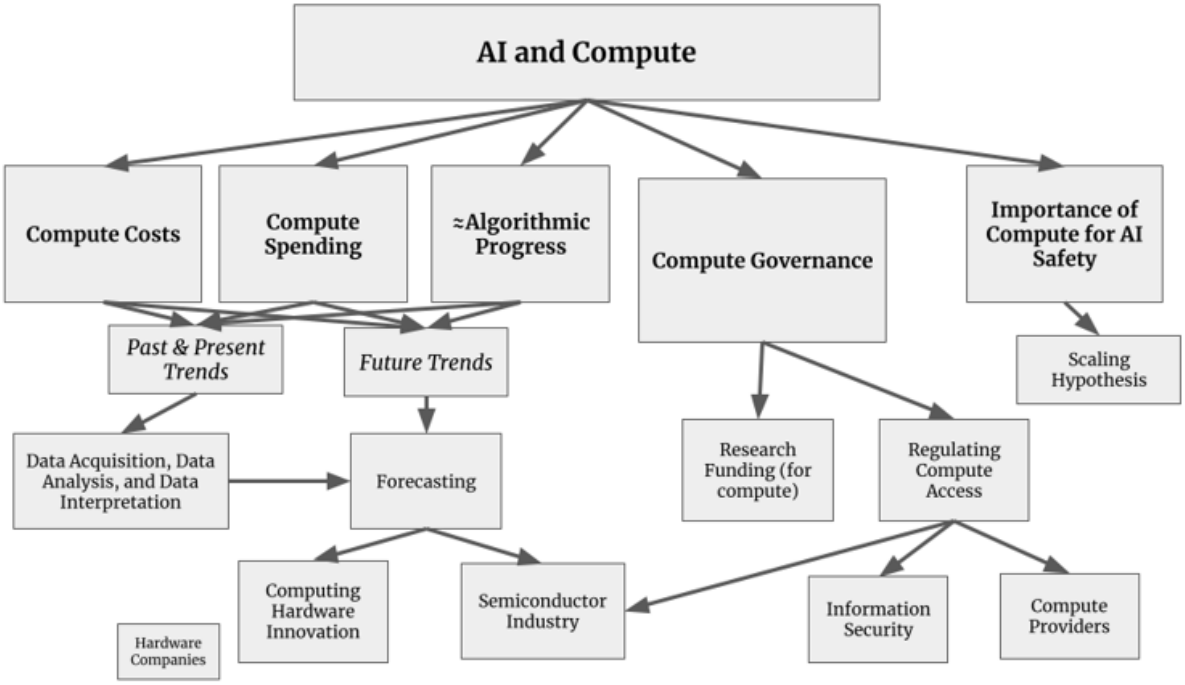


Figure A.1: Sketch of research domains for AI and Compute.

Compute Governance

- For a list of research questions, see "Some AI Governance Research Ideas" (Anderljung and Carlier 2021).
- What is the current budget of public research organizations for compute?
    - Should we lobby for an increase in compute budget?
        - If yes, how?
    - How does this compare to corporate AI research labs?

Scaling Hypothesis

- GPT-3 is a leading language model in terms of compute (≈3.64E+23) and parameters (≈1.75E+11). However, PanGu-α has more parameters (≈2.07E+11) but used less compute for training (≈5.83E+22) (estimates from here).
    - Would we expect better performance for additional training of PanGu-a? Why did they stop there?
    - What is the parameter-to-compute ratio for language models? What can we learn from it?
- New efficient architectures, such as EfficientNet, achieve similar performance while requiring less compute for training.

    - Are we expecting better performance if we scale up this architecture and increase the compute?

Compute Price

- How can we break up the AI and Compute trend? And what are the proportions of the relevant components, such as increased spending and more powerful hardware? ★
    - I think the current breakups of the compute trend into (a) more spending and (b) more performant hardware could be improved. My initial guess is that the performance of hardware is developing faster than Moore's Law. This is important as the limit is then often assumed when we hit 1% of GDP in spending. If (b) more performant hardware makes up a more significant proportion, this trend could be maintained for longer (at least from the spending side).
- How can we break up the price for compute? And what are the proportions of the relevant components?
    - I am discussing the operational costs in Appendix B. What is the ratio of the purchase price to energy, operations, engineering, etc?
- I have heard from researchers that the funding for compute is not the immediate problem, rather the required knowledge to use it and deploy the training run efficiently.
    - Does spending money on compute always entail hiring ML engineers? What is the fraction?

Compute Hardware

- Is there a "Learning Curve" for compute hardware (similar to solar power)?
    - Should we expect a strong effect with an increased spending in the future?
- Which emerging computing paradigms should we monitor? How can we prioritize among them? ★
    - In-memory computing, optical computing, hybrid approaches, 3D stacked chips, …
- Which metrics are the most informative for compute hardware performance? ★
    - Can we use benchmarks, such as MLCommons or Lambda Labs, for measuring hardware progress?
        - Can we based on those application-specific benchmarks create a benchmark like benchmark seconds per $?
    - Should we use metrics only in an application-specific context?
        - E.g., we could use the MLCommons benchmarks on language models to measure the performance of hardware for AI systems on language models.
    - How can we consider utilization in our metrics?
- Quantization and number representation: We have seen that neural networks can be quantized (going from 32bit float to 8bit integer representation) for inference without significant accuracy loss. This leads to substantial memory savings and inference speedups.
    - What is the limit of bit-width for training?
    - Which kind of speedup should we expect once we design dedicated hardware for the designed number representation? Integer representations require less energy and space on the die of the chip.
- Specialized architectures and ASICs

    - Why haven't we seen ASICs for AI systems yet? I outline some reasons in Section 4.2 but would like to dig deeper and do some forecasts. GPU and TPUs are still fairly general in regards to the computation they can execute.
    - Are there certain fundamental building blocks that we could significantly accelerate by designing specialized hardware for them?
        - E.g., could we design hardware that is significantly faster and more efficient for executing transformers?
    - We have seen technologies like bitcoin mining, or encryption, switching towards specialized chips or dedicated subprocessors.
        - What can we learn from those developments?
- Does narrow AI enable the potential for feedback loops for the design process of AI chips? E.g., (Mirhoseini et al. 2021).★
    - There are a couple of problems in the field of electronic design automation, such as place and route where AI could be highly beneficial and speedup the development process.
- Discontinuous progress
    - Which categories of AI hardware innovation could lead to discontinuities? How feasible are they?
    - Are the examples of discontinuous progress in computing capabilities for other applications, such as encryption, video processing, Bitcoin, or others? What happened there?
    - Are there some potential overhangs[1] within AI hardware domains (processing, memory, architectural, etc) which could be unlocked?
        - I think an innovation within the interconnect could unlock lots of hidden processing performance (discussed in Appendix B).
- Existing bottlenecks
    - What are existing bottlenecks in the computing hardware domain?
        - Can we distribute AI (training) workloads across multiple datacenters? What is the minimum bandwidth we need to maintain (in regards to the memory hierarchy: from on-chip, on-board, on-system, in-datacenter, internet, …)?
    - Which kind of speedup could we expect if we unlock the processor-memory performance gap?
- NVIDIA asks for a significant top-up for their commercial AI hardware. What is the price difference between consumer-grade GPUs and commercial ones?
    - Can we expect this to change in the near future when other competitors enter the market?
    - Are metrics based on consumer-grade GPUs a good metric for compute prices?
- Which companies will dominate the AI hardware market?
    - What can we learn from other technologies (e.g. encryption, Bitcoin, video accelerators) which resulted in their own chips?
- What is the time-to-market for compute hardware? Could this allow us to foresee upcoming developments?
- Assuming we acquire (e.g. stealing) the blueprints of transformative new computing technology or paradigm, can we just rebuild it?
    - What can we learn from the semiconductor industry? It seems super hard building fabs and the machines that build chips.
    - Is the knowledge residing in individuals? Can we just hire the staff? (E.g. China and TSMC in 2020.)

Compute Forecast

- For used compute trends: Which models should we include in the dataset? What are our inclusion criteria?
    - How do we deal with the more frequent emergence of efficient ML systems?
- Can rank lists, such as the Top 500 on supercomputers, provide insights on available compute?
    - How can we measure their compute performance for AI tasks?

Semiconductor Industry

CSET is already doing most of the work within this domain. I expect my questions could be answered by reading through all of their material.

- Which role has the semiconductor industry (SCI)?
    - Could AI capabilities be slowed down by manufacturing limitations (current semiconductor crisis)? Is it already slowed down? Should we expect more progress in the next decade?
    - What is the role of individual actors, such as TSMC or Nvidia in the compute hardware space?
        - Should we focus on regulating specific actors?
- How will the semiconductor industry develop in the next years?
    - What will be the impact of nations, such as the EU and the US, focusing on building a semiconductor industry?
    - Should we expect such development to decrease the price and increase research speed?

Algorithmic Efficiency

- Updating the AI and Efficiency trend with the newest data.
- Creating a similar dataset as AI and Efficiency for a different benchmark.
- New efficient AI systems are not optimized for hitting intermediate accuracy goals. We could tweak the efficient networks to achieve the accuracy benchmarks earlier and get better algorithmic improvements estimates.
- Learn more about the sample efficiency of AI systems.

AI Safety

- What is the role of compute hardware for AI safety?
    - Is there more to it than just enabling more compute? How can we use it in regards to AI safety?
- How can we limit access to potentially harmful AI systems via controlling hardware?
    - Can we remotely turn off hardware?
    - Can we integrate a self-destroy mechanism?
    - What is the role of low-level security exploits (such as Spectre and Meltdown)?
        - Could they be exploited to acquire compute of other actors? Or to make hardware unusable?

B. Metrics

Highlights

- The metric FLOPs/s is commonly used for discussing compute trends. However, as outlined in Section 1.1, the memory and interconnect are also of significance.
- FLOPs/s — which is often listed on specification sheets of hardware — does not give insights into real-world computing performance. We need to adjust this idealized factor by a utilization which is determined by various factors, such as the interconnect, memory capacity, and the computing setup.
- Consequently, it is important to base trends not purely on an increase in peak FLOPs/s but rather by effective FLOPs/s — measured by real-world performance.
- I suggest the investigation of benchmark metrics, such as the MLCommons or Lambda Labs benchmarks. These benchmarks provide insights into the performance of hardware with real-world workloads of different AI domains.

I have discussed various trends related to compute, and those trends often relied on metrics which we have then investigated over time. While metrics related to compute might initially seem more quantifiable than other AI inputs (such as talent, algorithms, data, etc.), I have encountered various caveats during my research. Nonetheless, I still think that compute has *the most quantifiable* metrics compared to the others.

In this Appendix B, I briefly want to present commonly used metrics, discuss some caveats, and propose some ideas to address those shortcomings.

B.1 Common used metrics for measuring hardware performance

The presented forecasting model for AI timelines is informed by one hardware-related metric: FLOPs/s per $. Algorithmic efficiency and money spent on compute are hardware independent. They are multiplicative with the FLOPs per $ (see Section 4.1).

We have the following options to *make progress* or *acquire more compute*. Assuming all the other metrics are constant, we can either:

- Increase the computing performance [usually measured in FLOPs/s]
  - By improving one of the listed components: logic, memory, or interconnect (see Section 1.1)
- Decrease the price [$][2]
- Decrease the energy usage [W] which decreases the price over time [$]

While metrics, such as the price and energy usage, are more clearly defined and measurable, the computing performance *is more than just FLOPs/s* and depends on various factors which I will discuss.

I will present metrics connected to the three basic components: logic, memory, and interconnect.

Logic

FLOP (with its plural FLOPs)

Presented in Section 1, our atomic entity is the operation. FLOP refers to a floating point operation which is a type of operation on a specific number representation, a floating point number. Nowadays, this term is commonly used interchangeably, independent of the number representations.

However, it is of importance which number representation is used, as this defines the performance of the hardware. Commonly used in Machine Learning are float16, float32, bloat16, int8, and int16. For an introduction to number representations see here or here.

Also commonly used is Petaflop/s-day. It's also a quantity of operations. A petaflop/s is 10^{15} floating point operations per second for one day. A day has

$84,400$ seconds $\approx 10^{15}$. That makes 10^{20} FLOPs.

FLOPS or FLOPs/s

A quantity of operations does not tell us *how long it will take* to process them. For a given quantity of operations (*X* FLOPs) executed in a given time, we have the metric: FLOPs per second (FLOPS or FLOPs/s). This is the most common metric used for the processing power of hardware or whole datacenters (e.g., see the Top 500 List).

FLOP/s per $

If we now take the purchase price of the hardware into considerations, we can now divide the computing performance in FLOPs/s by the purchase price to learn about FLOPs/s per $. This puts different hardware into comparison and tells how many operations specific computing hardware can execute per second for a single dollar.[3]

Note that this metric commonly ignores the operation costs of hardware that is partially defined by FLOPs/s per Watt.

FLOPs/s per Watt

FLOPs/s per Watt is similar to FLOPs/s per $. However, instead of the monetary costs it considers the energy costs. It describes the efficiency of the hardware.

Energy efficiency is a key metric in computer engineering, as hardware is often limited by its heat dissipation (see power wall). A higher energy efficiency often allows us to increase the clock frequency or other components which then allows us to compute more FLOPs/s.

The operation costs — here the energy cost — is often excluded in FLOPs/s per $ estimates[4], as the initial purchase price makes up most of the cost.[5]

However, I think that considering all the other operational costs, such as cooling, system setup, system administration, etc, might change this picture and would be interested in estimates.

Memory

Bytes

The memory capacity is described in Bytes or Bits and is familiar to most people. Most of the confusion is usually around the different types of memory: cache, on-board memory, RAM, HDD, SDD, etc.

One of the most important to concepts to understand is memory hierarchy. It separates different storage types in regards to their response time, bandwidth and capacity. The heuristic: low response time [in seconds] and high bandwidth [in Byte per second] leads to lower memory capacities.

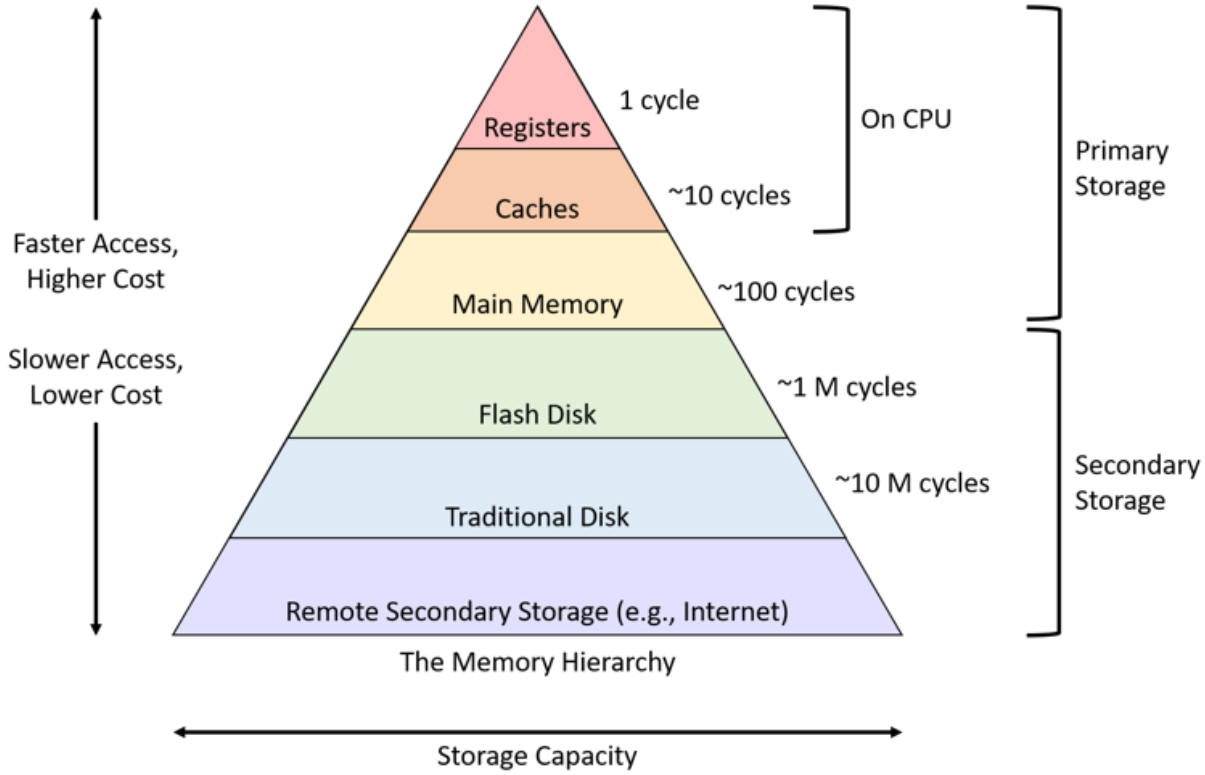


Figure B.1: Memory hierarchy in computing systems (Source).

Why is the memory capacity of importance for the performance of the system? As already discussed in Section 1.1, your logic can be bottlenecked if it does not get supplied with enough data. Therefore, data locality is important. If you are training an AI system, you would like your weights and parameters to be as local as possible to enable fast read and write access. Therefore the onboard memory of your GPU is often important (as a minimum you should fit all of your parameters on it). Whereas it is theoretically possible to offload them to other memory systems, this comes with a significant increase in training time and makes it usually unfeasible.

Interconnect

We have introduced metrics related to the logic and the memory. To now move data from the memory to be processed in the logic element, we require an interconnect.

Bytes/s

The memory bandwidth is described in Bytes per second, short B/s. As previously described, this memory bandwidth depends highly on the locality. Moving data from the on-board (e.g. called GPU memory) to be processed in the logic is orders of magnitude faster than, for example, if you need to communicate data from local on-board memory, to a different board via various interconnects.[6]

Traversed edges per second (TEPS)

To address the shortcomings of only measuring the computing capabilities of the logic elements (more in B.2), the measure “traversed edges per second (TEPS)” was introduced. It is a measure of the interconnect capabilities and computational performance.

This is an important measure, especially for high performance computing, as those datacenter consist of hundreds of individual computers and data needs to be moved around from computer to computer — communication through various levels of memory hierarchy.

The Graph500 is a rating based on this metric. AI Impacts also wrote a post on the cost of TEPS.

This is just a brief selection of relevant metrics. I’ve focused on those that I have seen commonly used in my references.

B.2 Some caveats of common used metrics

This section discusses some shortcoming of the presented metrics. I list the caveats in order of my felt importance (from more important to less important).

Utilization: what happens in reality

The FLOP/s on datasheets present peak performances of the hardware — what the hardware is able to process, assuming a perfectly balanced workload. Nonetheless, this is rarely the case in the real world. Our hardware is often only processing X FLOP/s of the theoretical achievable Y FLOPs/s. We refer to this as the utilization: $\frac{X}{Y}$.

The achieved utilization depends on various factors, I briefly discuss communication, parallelism, and software.

Also, we discuss some ideas around utilization in an upcoming piece on estimating compute (here it is).

Communication

> "*A supercomputer is a device for turning computational complexity into communication complexity.*"

— Beth's commented quote on Crox's post.

I have discussed in B.1 Interconnect, and Section 1, that the interconnect is crucial to the operations of the logic. If the logic is not supplied with enough data, or the interconnect cannot write the results fast enough, the logic is bottlenecked by the interconnect. This behavior is not captured by peak FLOPs/s.

How big of a deal is this? I think it’s a big deal. The memory-processor gap is well known within the computing community (see this, the roofline model, memory-bound, or compute-bound).

An example is the specifications from NVIDIA's V100[7] to A100[8]. The A100 is the succeeding generation. This generation steps focused next to doubling the tensor performance, also on doubling the on-board bandwidth but also the off-board interconnect. Whereas, not focusing on the FP32 or FP64 performance, only increasing it marginally.

Parallelism

AI hardware builds on parallelism on various layers. Parallelism in the chip architecture, but also up to having multiple computing systems where we distribute the workload.

However, for this we require a workload which is parallelizable. Most AI workloads are highly parallelizable and this is the reason why we switched from CPU to GPUs/TPUs. However, not every workload is perfectly adapted for the underlying hardware. For this reason the peak performance is rarely achieved, as all the components (such as network architecture, hyperparameters, etc.) would need to be matched to the underlying hardware.

This also means, 10 GPUs with each 10 TFLOPs/s do not equal 100 TFLOP/s. The parallelizability and the communication overhead reduces the achieved performance and tweaking the workload and distribution is necessary.

OpenAI discusses some ideas around parallelism in the piece: "An Empirical Model of Large-Batch Training".

Software

Software, such as compilers and libraries, helps us to translate our high-level languages into actual machine-readable code. This machine-readable code should then leverage the features of the hardware architecture, such as the parallel architecture. Examples of this are CUDA, Triton, or XLA. Ideally, this software helps with the above-outlined problem of parallelism and communication. However, this is hard and some problems are NP-hard or NP-complete.[9] Consequently, the peak performance is rarely achieved.

The number representation

Talking about the amount of FLOPs or FLOP/s without mentioning the number representation gives limited insights. Especially the bit width, such as float16, or float32, is important. Short bit-widths can be processed faster and also moved around faster round — as the corresponding in- and outputs are smaller in size. Already now, we see that integer processing comes with significant speedup and memory savings (but is mostly used for inference).[10] For the future, I am assuming that we see more specialized number representations such as the bfloat from Google, emerging in the future.

Therefore, I would also criticize the term FLOP, as non-floating operations are already present. However, the term seems now to be used interchangeably with OPs (even though it is technically not correct).

Consequently, I think it is fair to say that we currently have a *processing overhang*. I refer to the capabilities of the logic — we can currently theoretically process more FLOPs per second than we can read and write given by our interconnect or memory.

Therefore, improving the memory bandwidth —without increasing the processing performance— will lead to more computing capabilities.

Operations costs (energy, infrastructure, etc.)

For the moment, most estimates neglected the operations costs, such as energy and infrastructure. I think this is fair —at least in regards to energy— given some back-off-the-envelope calculations. Nonetheless, we should continue monitoring this. With potential longer innovation cycles and an economy of scale (discussed in Section 4.2), the purchase price might decrease and the operations costs might become a more significant proportion.

Additionally, I think engineering costs might not be neglectable (running a computing farm, setting it up, etc.). I'd be interested in estimates. To get a more complete picture, one could get some insights into those costs by estimating the costs of FLOPs/s per $ from cloud computing providers, such as AWS or Google Cloud (minus their premium for profit).

What is an operation?

We describe a FLOP or OP as the atomic instruction which we count. However, on the computer architecture level this usually gets broken into multiple instructions — how many and what they are like depends on the instruction set architecture. This is the assembler code. This type of code is closer to the hardware and could provide us more reliable insights into the performance with metrics, such as instructions per cycle. However, this would require more work and gets into the weeds of computer engineering.

I think for this reason, FLOPs and multiply-accumulate operations (MACCs) often get conflated. Some use them interchangeably and assume one FLOP equals one MACC, whereas others assume a MACC equals two FLOPs.

In the end, a MACC is hardware-specific operations. If the underlying hardware consists of a fused-multiply-add unit, then a MACC is nearly as costly (in terms of latency) as a single FLOP. However, if it does not or the operations (multiply, and then addition) are not done consecutively, it equals two FLOPs.

Nonetheless, I don't want to go into the details here. But assuming we got all of the above-mentioned caveats right, our metrics still have some limitations. I do not think that this is of importance given our current trend line where we see a doubling in training compute every six months. However, for my previous research on ML on the edge (optimize AI systems for the deployment on small resource-constrained embedded devices.) where our error bars need to be smaller than one order of magnitude, this played an important role. Additionally, often people get confused when they make their own estimates, and then numbers do not add up.

B.3 Concluding thoughts

I have listed various caveats on commonly used metrics and under which circumstances they are of more or less use. Measuring AI hardware does not rely on a single metric that gives you *all the information* but that is rarely the case for any domain. Overall, I am giving the recommendation to adjust hardware performances

with utilization rates and monitor developments in the memory bandwidth more closely.

We are working on a piece with more insights on the utilizations and some advice on how to estimate training compute and the connected utilization of the system (*link to be added by the end of 2021; ping me if not*).

I'm also expecting that measuring those metrics will get more complicated over time due to the emergence of more heterogeneous architectures, and we might see more specialized architectures for different AI workloads.

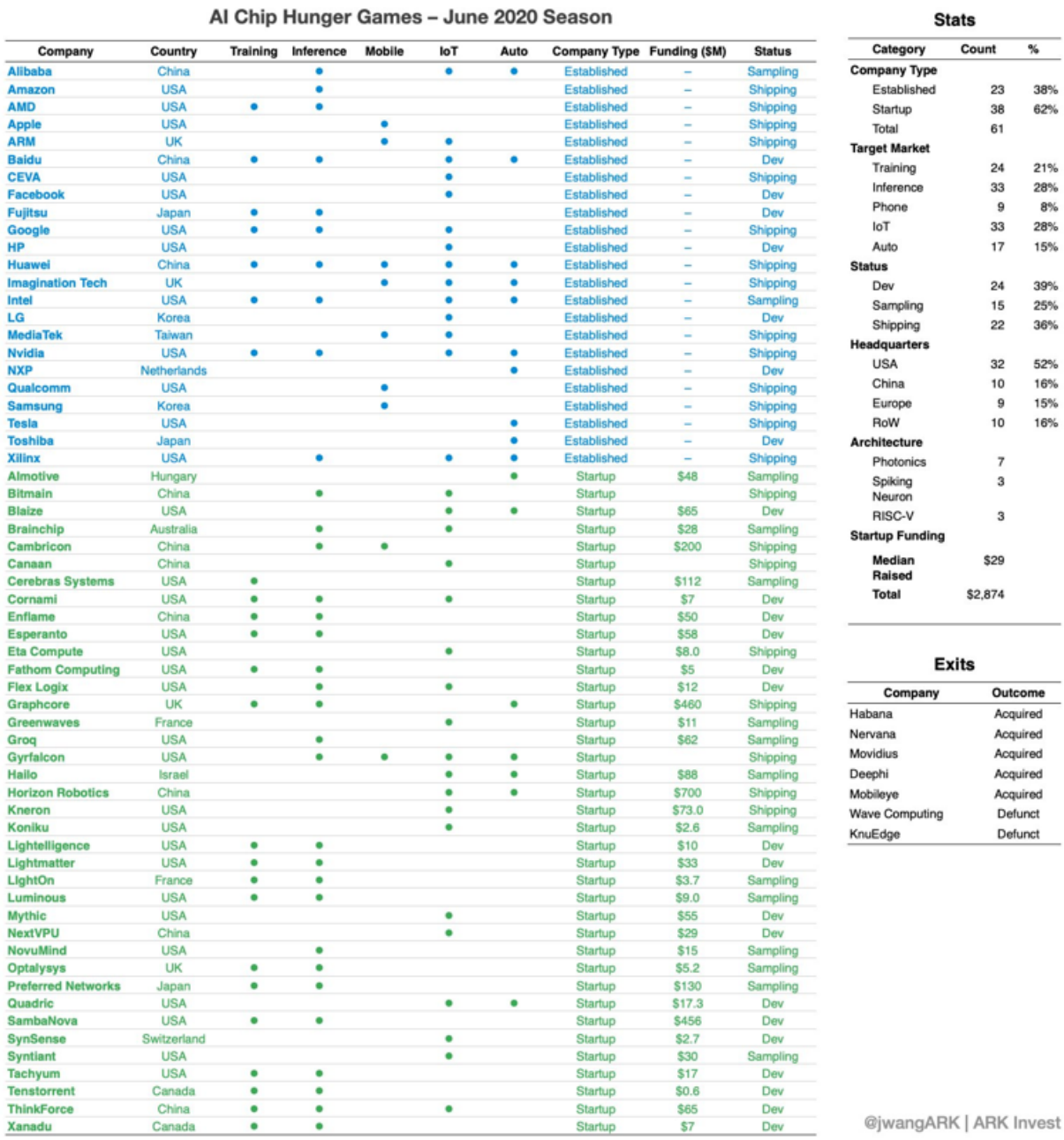
Consequently, I am interested in metrics that are workload-dependent and can give us some insights into the performance of AI hardware.
MLCommons is a project which lets hardware manufactures published their training times for different AI domains. There we find training times for different hardware, hardware setups, and different AI workloads: e.g, image recognition and NLP. Analyzing this data would allow us to analyze trends based on the training time which already encapsulates listed caveats such as the utilization, workload-dependent parallelizability, and memory access. For example, we could calculate performance gains for real-world workloads of hardware over time (unfortunately, the data only goes back to 2017). Additionally, we get more insights into closed systems such as Google's TPU.

I have made this data available in this sheet and would be interested in some analysis.

The GPU data from Lambda Labs is another promising dataset.

C. AI Hardware Startups

This is a list of AI hardware startups that are working on new types of hardware paradigms, often optical computing (as a hybrid approach between digital and optical).



AI Chip Hunger Games – June 2020 Season

Company	Country	Training	Inference	Mobile	IoT	Auto	Company Type	Funding ($M)	Status
Alibaba	China		●		●	●	Established	–	Sampling
Amazon	USA		●				Established	–	Shipping
AMD	USA	●	●				Established	–	Shipping
Apple	USA			●			Established	–	Shipping
ARM	UK			●	●		Established	–	Shipping
Baidu	China	●	●		●	●	Established	–	Dev
CEVA	USA				●		Established	–	Shipping
Facebook	USA				●		Established	–	Dev
Fujitsu	Japan	●	●				Established	–	Dev
Google	USA	●	●		●		Established	–	Shipping
HP	USA				●		Established	–	Dev
Huawei	China	●	●	●	●	●	Established	–	Shipping
Imagination Tech	UK			●	●	●	Established	–	Shipping
Intel	USA	●	●		●	●	Established	–	Sampling
LG	Korea				●		Established	–	Dev
MediaTek	Taiwan			●	●		Established	–	Shipping
Nvidia	USA	●	●		●	●	Established	–	Shipping
NXP	Netherlands					●	Established	–	Dev
Qualcomm	USA			●			Established	–	Shipping
Samsung	Korea			●			Established	–	Shipping
Tesla	USA					●	Established	–	Shipping
Toshiba	Japan					●	Established	–	Dev
Xilinx	USA		●		●	●	Established	–	Shipping
Almotive	Hungary					●	Startup	$48	Sampling
Bitmain	China		●		●		Startup		Shipping
Blaize	USA				●	●	Startup	$65	Dev
Brainchip	Australia		●		●		Startup	$28	Sampling
Cambricon	China		●	●			Startup	$200	Shipping
Canaan	China				●		Startup		Shipping
Cerebras Systems	USA	●					Startup	$112	Sampling
Cornami	USA	●	●		●		Startup	$7	Dev
Enflame	China	●	●				Startup	$50	Dev
Esperanto	USA	●	●				Startup	$58	Dev
Eta Compute	USA				●		Startup	$8.0	Shipping
Fathom Computing	USA	●	●				Startup	$5	Dev
Flex Logix	USA		●		●		Startup	$12	Dev
Graphcore	UK	●	●			●	Startup	$460	Shipping
Greenwaves	France				●		Startup	$11	Sampling
Groq	USA		●				Startup	$62	Sampling
Gyrfalcon	USA		●	●	●	●	Startup		Shipping
Hailo	Israel				●	●	Startup	$88	Sampling
Horizon Robotics	China				●	●	Startup	$700	Shipping
Kneron	USA				●		Startup	$73.0	Shipping
Koniku	USA				●		Startup	$2.6	Sampling
Lightelligence	USA	●	●				Startup	$10	Dev
Lightmatter	USA	●	●				Startup	$33	Dev
LightOn	France	●	●				Startup	$3.7	Sampling
Luminous	USA	●	●				Startup	$9.0	Sampling
Mythic	USA				●		Startup	$55	Dev
NextVPU	China				●		Startup	$29	Dev
NovuMind	USA		●				Startup	$15	Sampling
Optalysys	UK	●	●				Startup	$5.2	Sampling
Preferred Networks	Japan	●	●				Startup	$130	Sampling
Quadric	USA				●	●	Startup	$17.3	Dev
SambaNova	USA	●	●				Startup	$456	Dev
SynSense	Switzerland				●		Startup	$2.7	Dev
Syntiant	USA				●		Startup	$30	Sampling
Tachyum	USA	●	●				Startup	$17	Dev
Tenstorrent	Canada	●	●				Startup	$0.6	Dev
ThinkForce	China	●	●		●		Startup	$65	Dev
Xanadu	Canada	●	●				Startup	$7	Dev

Stats

Category	Count	%
Company Type		
Established	23	38%
Startup	38	62%
Total	61	
Target Market		
Training	24	21%
Inference	33	28%
Phone	9	8%
IoT	33	28%
Auto	17	15%
Status		
Dev	24	39%
Sampling	15	25%
Shipping	22	36%
Headquarters		
USA	32	52%
China	10	16%
Europe	9	15%
RoW	10	16%
Architecture		
Photonics	7	
Spiking Neuron	3	
RISC-V	3	
Startup Funding		
Median Raised	$29	
Total	$2,874	

Exits

Company	Outcome
Habana	Acquired
Nervana	Acquired
Movidius	Acquired
Deephi	Acquired
Mobileye	Acquired
Wave Computing	Defunct
KnuEdge	Defunct

@jwangARK | ARK Invest

Figure C.1: Overview from June 2020 by @draecomino (Tweet Source).

List of AI hardware startups I stumbled upon during my research:

- Fathom Radiant
- Luminous Computing
- Rain Neuromorphics
- Optalysys
- LightOn
- LookDynamics
- Cambricon Technologies
- Graphcore
- Mythic AI

- Lightmatter
- Cerebras
- Quantum Computing
  - List of quantum computing companies by Christopher Phenicie
  - Dwave

Acknowledgments

You can find the acknowledgments in the summary.

References

The references are listed in the summary.

---

1. An overhang refers to a situation where large amounts of resources that are already available cannot be used yet. If it is resolved, large amounts are unlocked immediately (Dafoe 2018). ↩

2. As already discussed in Section 4.2, even when hardware does not significantly increase in computing performance, the price can still decrease significantly due to longer R&D cycles and an economy of scale. ↩

3. See this Wikipedia List or this post by AI Impacts for historic trends of FLOPs/s per $. ↩

4. For example in our discussed timeline forecast, this Wikipedia List, or this post by AI Impacts. ↩

5. A quick back-off-the-envelope calculations: A NVIDIA A100 consumes around 6.5kW at peak usage. Assuming 0.12$ per kWh, it costs around 6,800$ per year for running this hardware. This is rather neglectable to the purchase price, given that an NVIDIA A100 costs around $200,000 to $300,000. ↩

6. For an NVIDIA A100, the on-board memory bandwidth is around 2GB/s, whereas interconnect with additional A100's using NVIDIA'S specialized NVLINK, one achieves up too 600 GB/s. And only 64GB/s using the standard PCIe Gen4 interface (see this datasheet). ↩

7. NVIDIA V100 datasheet ↩

8. NVIDIA A100 datasheet ↩

9. Class on Advanced Compiler Optimization ↩

10. Integer representation (instead of floating point), saves energy and requires less space on the chip die. See Computing's Energy Problem - Slides and Computing's energy problem (and what we can do about it) ↩