An exploration of optimization techniques in artificial intelligence microchip design

Research question: How can artificial intelligence chips be optimized for both performance and energy efficiency?

Subject: Computer Science

Word count: 3663

Table of Contents

Introduction 3

Background 4

Optimization Techniques 5

Case Study 1: Apple Neural Engine 10

Case Study 2: Google's TPUs …….. 11

Case Study 3: NVIDIA's GPUs ……... 12

Conclusion ……... 13

Bibliography …… 16

Introduction

Artificial intelligence, or AI, has permeated almost every facet of our everyday lives, from personalized recommendations on streaming platforms and virtual assistants to self-driving cars and finance. As AI continues to be marketed towards larger amounts of people and process enormous amounts of data, its computational requirements are increasing at an alarming rate. Due to the complexities of the problems AI is used to solve, the hardware AI models run on must also improve. However, the current hardware we use is not enough to handle the computing power and speed required by AI algorithms, sparking the need for AI chips, specialized hardware to meet the computational demands of AI tasks. While AI chips greatly improve processing speed, their energy demands and cost increase when raising efficiency. Therefore, optimizing AI chips is not only about their efficiency, but also about balancing efficiency with their massive energy demands and costs.

In technology, a chip refers to a microchip, a microscopic integrated circuit unit with semiconductor components called transistors. Semiconductors are materials that can control electrical currents, making them essentials for most modern electronics. Transistors are a type of semiconductor that are used to control currents by acting as a switch or an amplifier and

control logic or memory in microchips (Urwin). AI chips mostly deal with the logic side, handling intensive data processing needs, a task beyond the capabilities of traditional general-purpose chips like central processing units (CPUs). However, unlike CPUs which mostly handle sequential processing, processing chips such as graphics processing units (GPUs), application specific integrated circuits (ASICs), and tensor processing units (TPUs) are capable of both processing large datasets and conducting operations in parallel.

Background

AI algorithms, neural networks in particular, require millions of simultaneous mathematical operations. Originally designed to run graphics, general-purpose GPUs (GPGPUs) shine in this aspect, as they use thousands of cores to efficiently perform operations in parallel, reducing the time required to processive enormous sets of data. Parallel processing is especially useful when applied to tasks where the same mathematical operation is being performed on multiple data points simultaneously such as in convolutional neural networks (CNNs). Developed by Google, TPUs are application-specific processors optimized for tensor operations, computations using multi-dimensional arrays, which are an integral part of AI tasks (Bigelow). Unlike GPUs which are general purpose, ASICs are chips that are built for specific tasks, giving them optimal efficiency and performance such as image recognition or language processing (“What Is ASIC”). Each type of chip has its own strengths and weaknesses, making them more efficient for different uses. While GPUs provide flexibility in tasks, ASICs and TPUs offer efficiency for specialized operations.

There are numerous metrics to define efficiency in AI chips, such as Operations per Watt (OP/W), latency and throughput, and energy per inference. OP/W describes the number of operations performed per unit of energy used. A higher OP/W indicates a higher energy efficiency. Latency is a measure of the time taken to complete a single task, while throughput is

a measure of how many tasks can be processed in a certain amount of time. Both latency and throughput are important for AI applications in real-time, such as autonomous driving, since it is necessary for the AI model to make decisions quickly. Inferencing is when AI models use the data they were trained on to recognize patterns in data it has not seen before. Thus, energy per inference measures the energy required to process one input, such as classifying an image as an apple or orange. These different metrics are critical when evaluating the speed and energy efficiency of AI chips.

Many challenges exist when attempting to optimize AI chips through trade offs between performance, energy consumption, and cost. When performing many computations and operations, high-performing AI chips generate a significant amount of heat, requiring advanced cooling methods to prevent thermal throttling—when the chip begins to sacrifice power to reduce its temperature (Abidi). Optimizing AI chips for speed or power usually comes at the price of a much higher energy consumption, making it difficult to find an equilibrium between performance and energy consumption. Additionally, as AI models will only grow larger, optimizing AI chips for scalability without compromising on their efficiency will become increasingly more difficult. Addressing all these challenges requires a multitude of trade offs between different the different efficiency standards.

Optimization Techniques

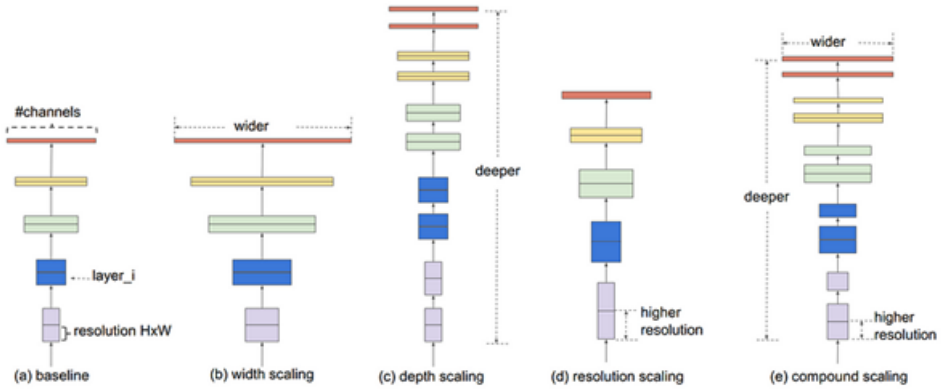
AI chips can be optimized through multiple different viewpoints. These include innovation on the hardware side, the software side, or on the algorithmic side. Hardware optimizations include changing the aspects of the materials used and using different types of chips. For instance, shrinking the size of transistors increases the number that can be used, allowing greater OP/W. Three-dimensional chip stacking is a technology in which semiconductor materials are placed vertically on top of each other to form a single integrated circuit. 3D chip

stacking allows for a higher component density within a smaller space and for improved performance and lessened power consumption due to shorter connections in between the components (“Understanding 3D Chip Stacking”). However, this technique also presents challenges such as high temperature and cost, needing further cooling technology. Normally, with single core chips, operations must be performed sequentially, but multi-core chips allow many processing units to perform independent operations in parallel, greatly boosting speed (Swain et al). The advent of ASIC technology has drastically enhanced the speed at which chips can execute specific tasks. The ability for a chip to be designed for a particular function allows it to compute operations much faster than a general-purpose processor that is built to handle a wide variety of tasks. Since ASICs focus on certain operations, they outperform GPUs in terms of energy efficiency, making them work ideally for large scale AI processes like natural language processing. Edge AI chips prioritize efficiency on the local level of a consumer device, allowing for real-time efficiency on devices like phones. These chips are optimized to save power, allowing AI models to run without relying on cloud AI (“What Is Edge AI?”). Because these chips are local to the device the AI is running on, data transfer is reduced, reducing energy consumption.

Software optimization techniques mainly focus on improving the use of computational resources during AI tasks. These techniques optimize data usage and eliminate redundant computations to increase the speed of AI processes. Model pruning is a technique in which less contributing parameters in AI models are removed, reducing both the size and the complexity of machine learning models (Howard et al). However, when pruning parameters, the accuracy of the model is bound to be reduced, so it is necessary to be careful about which parameters to remove to have the least significant impact on the accuracy of the model. For example, within neural networks, there are individual weights at each node for turning an input into an output. Model pruning could be targeting individual weights or removing entire layers. Furthermore,

pruned models are smaller, allowing for less memory for storage, reducing energy consumption. An example of model pruning in the real world is the MobileNet model produced by researchers at Google. The MobileNet model had only 4.2 million parameters, while the VGG 16 model had 138 million. However, the MobileNet model was able to be significantly close in accuracy, as it had a 70.6% accuracy, while the VGG 16 had a 71.5% accuracy (Howard et al). This difference demonstrates the cost in model pruning, but also an effective instance of it, allowing for minor decreases in accuracy, but major increases in energy efficiency. Quantization is a software technique that improves efficiency by reducing how precise numerical data are represented in machine learning models (Howard et al). Neural networks usually work with 32-bit numbers called floating-point numbers, which require a lot of memory resources to be precise, but quantization converts these 32-bit floating-point numbers into lower precision representations such as 16 or 8-bit integers, significantly reducing the amount of memory and computational resources used. There are many methods to implement quantization, including dynamic, static, and quantization aware training (QAT). The dynamic approach to quantization adjusts the precision of data while the model is being run, reaching a balance between efficiency and accuracy. However, static quantization sets a fixed precision throughout the entire model, allowing for less energy consumption, but at a possible cost of accuracy. QAT implements quantization while the AI model is being trained on the training dataset, making the model capable of adapting to reduced precision in data and minimizing the resulting decrease in accuracy (Zafrir et al). Within Google's TensorFlow framework, quantization techniques have optimized models like BERT for natural language processing. BERT operations were quantized to be 8 bits for a variety of tasks with minimum loss in accuracy, decreasing its memory footprint. As shown in the table below, when the BERT model was quantized using the QAT approach, the accuracy either decreased by minimal amounts, or in some cases even increased or stayed the same.

Algorithmic optimization techniques instead focus on the AI model, making it more compatible and easier to run on a chip. These techniques reduce the number and complexity of computations needed during training and further inferencing, increasing the speed of the model. Neural networks use multi-layer calculations to produce an output given some input, requiring many computations, with convolutional neural networks being an advanced, complex version. However, neural networks such as MobileNet and EfficientNet have been designed specifically to optimize performance while minimizing the complexity of operations. This is achieved through the use of techniques like depthwise separable convolutions and reducing layer complexity. MobileNet, for example, uses depthwise separable convolutions, which splits convolution operations into depthwise convolution and pointwise convolution (Howard et al). The separating of these complex operations into two simple ones allows for the significant reduction of parameters and necessary computational resources, making the model faster. The EfficientNet model, however, uses a technique called compound coefficient which scales models up effectively by balancing the scale in the three dimensions of width, depth, and image resolution instead of focusing on only one dimension, as seen in the figure below (Sarkar).



These models are valuable when paired with devices with limited hardware resources, such as in edge devices like smartphones. However, efficient neural networks come with many tradeoffs, as simplifying parameters or operations to reduce the complexity of a task can often lead to a

decrease in accuracy, however minor it may be, which is never ideal. Especially when using models for tasks with needs for precise outputs, these simplified operations may not produce a useful output due to low precision. Additionally, fine-tuning what parameters or operations are necessary takes a lot of time and computational resources, using more energy simply for training before its actual use. Knowledge distillation is a technique in which larger, more complex “teacher” models train smaller, more efficient “student” models to mimic their behavior (Zedeh et al). This technique involves training the student model to reproduce the outputs of the teacher model, allowing for precision expected from a complex model to be produced in a smaller and faster model. This is extremely useful when making AI models compatible when paired with devices with limited hardware like phones. This technique is used to train smaller versions of the BERT model where the student model adopts a different network for the student to learn the contextual dependence of certain tokens (words) from BERT’s outputs (Zadeh et al). An advantage to knowledge distillation is that it reduces the amount of required training data, as the student model learns from the outputs of the teacher model instead of from raw datasets. However, knowledge distillation also has weaknesses, as the student model’s accuracy is extremely dependent on the accuracy of its teacher model, maintaining the need for an original complex model, requiring many resources. Furthermore, the student model may not encapsulate the entirety of the teacher model’s complexity, sacrificing accuracy for efficiency. Therefore, much experimentation is needed to fine-tune how a student model learns from its teacher model.

Case Study 1: Google’s Tensor Processing Unit (TPU)

When it was first announced in 2016, Google’s TPU was a groundbreaking innovation designed to accelerate specific machine learning operations like neural network processes and

deep learning. The TPU was mainly focused on matrix multiplication, a complex mathematical operation used in AI models that facilitates the handling of large datasets. Unlike the general-purpose CPUs and GPUs, TPUs are a type of ASIC, that are tailored to the demands of neural networks intense computational requirements, making significant improvements in efficiency. The architecture behind Google's TPU mainly focuses on achieving a high throughput (number of tasks in a certain time) and energy efficiency. Originally, general-purpose graphics processing units (GPGPUs) were the standard for machine learning tasks due to their ability to conduct thousands of operations in parallel, but TPUs have achieved a performance of up to 30-80 times better per watt for specific AI operations. In a study, it was found that the TPU was able to achieve 3.5 times the memory capacity of the K80 GPU while decreasing energy consumption by more than half (Jouppi et al 2017). This clearly exemplifies the ability for TPUs to be both faster and more energy efficient. TPUs have been integrated into many parts of Google's large-scale applications like image recognition in Google Photos and natural language processing in Google Translate, immensely reducing computational demands. Latency (time taken to complete a task) is necessary when marketing these applications to consumers to produce the most positive customer feedback. TPUs excel in this task, as they can produce inferences at a much higher rate with low energy consumption. With the development of the TPU v4, Google has managed to use 2-6 times less energy and produce 20 times less CO_2 than other modern chips (Jouppi et al 2023). With Google integrating TPUs in many of its data centers for many different types of AI applications, the potential for TPUs to transform AI processing has been revealed. TPUs not only achieve high performance goals, but they also meet sustainability goals through the reduction of energy usage. As AI models will only continue to be scaled upwards and adopt even more niche applications, TPUs exist as a sustainable path for efficient AI chips.

Case Study 2: Apple Neural Engine (ANE)

The Apple Neural Engine (ANE) was introduced with the innovation of the A11 bionic chip in 2017 which was implemented into their iPhone products. The ANE is a type of neural processing unit (NPU), a specialized microprocessor that is designed to mimic the human brain. The chip represents the company's focus for edge AI processing as their goal is to have this chip in edge devices for their consumers rather than in their data servers that consumers can access through the cloud. The ANE is different from TPUs, as instead of focusing on large data operations, it focuses on quickly delivering efficient AI capability to mobile devices. The ANE already powers multiple features such as facial identification, augmented reality, and computational photography—photographs that use computation instead of optical processes—in Apple's iPhones and iPads (Zubarev). By integrating the ANE into their mobile devices and reducing reliance on cloud processing, Apple has made both speed and energy efficiency improve. The ANE uses specialized hardware for the specific tasks it is use for like image and vice recognition. Like Google's TPUs, the hardware accelerators are optimized with matrix operations in mind, allowing for faster computations with lower energy consumption when compared with traditional CPUs and GPUs. Furthermore, the ANE uses dynamic power scaling, a technique in which energy consumption is adjusted based on the task, making sure energy is not wasted. The ANE's on-device capabilities extend beyond ease of use and efficiency, as privacy and security are also enhanced due to limiting the amount of data transfers to external servers. When making AI chips for consumer use, the user experience is extremely important, especially for companies like Google and Apple that market their devices to large audiences. Power consumption is a main issue that is addressed by these companies to improve battery life on their devices and their device performance.

Case Study 3: NVIDIA's GPUs

NVIDIA has been a leader in the GPU development scene since gaining a foothold in the gaming industry with the GeForce 256 in 1999. While their GPUs were initially designed to run computer game graphics, they shifted towards AI processes, becoming the modern world's leading AI chip company. NVIDIA's GPUs, like the A100 Tensor Core are able to find an equilibrium between performance and energy cost, which, combined with GPUs' general-purpose capability, makes them useful for many different large-scale AI applications. NVIDIA's GPUs utilize tensor cores, specifically designed GPU cores that enable dynamic calculations and computing with varied precision ("CUDA Cores vs Tensor Cores"). The use of these tensor cores enables NVIDIA's GPUs to achieve a greater throughput than more traditional GPUs as shown in the graph below comparing the DGX A100 that uses eight A100 GPUs against standard chips. The use of the A100 GPUs increases performance for AI training up to six times, while the AI inferencing performance is about 172 times as much as a traditional CPU (*NVIDIA A100 Tensor Core*).

Additionally, NVIDIA has developed a toolkit called CUDA, a software application that allows AI software developers to optimize their models based on the hardware NVIDIA provides, enhancing compatibility and furthering efficiency (Oh). Today, virtually every major technological company utlizies NVIDIA's AI chips, including Amazon, Google, and Meta, with their biggest buyer, Microsoft, buying 458,000 NVIDIA GPUs in 2024 for $31 billion (Okemwa). With NVIDIA's lead in GPU development and their focus on increasing the usability of their chips exemplified by their assiting software toolkits, it becomes clear why this is the case. NVIDIA's GPUs demonstrate the need for advanced innovations in chip design to keep up with the rapidly advancing complex software.

These three case studies of innovations in AI focused chip design illustrate the strategies companies use to optimize AI chip efficiency. Google's TPUs demonstrate the power of ASICs in large-scale applications, while the ANE demonstrates the potential capability for on-device AI applications. The case of NVIDIA's GPUs highlights the importance of general purpose hardware like GPUs to enable a wide variety of AI applications by balancing performance and flexibility. Overall, these examples showcase the necessity of tailoring AI chips to specific AI tasks, as well as the importance of maintaining hardware and software compatibility.

Conclusion

Optimizing AI chips requires the consideration of multiple aspects, including cost, energy efficiency, and performance, which often come at the expense of the others. Optimizing these chips happens both on the hardware side and the software side, demanding a delicate approach from each to accommodate the other. While hardware optimizations like innovative chip architectures with smaller transistors and 3D stacking allow chips to improve efficiency by increasing OP/W and overall performance, hardware developments are extremely expensive, requiring massive amounts of time, expertise, and resources. Big tech companies such as Google, Apple, and NVIDIA have spent billions of dollars refining their chips, which is a barrier to smaller companies without those resources. Many companies are then reliant on the major chip developer, NVIDIA, to obtain AI hardware, resulting in concerns about an AI chip monopoly. On the other hand, software techniques are cheaper and more flexible, allowing models to better fit the chips they run on, maximizing the performance of the hardware. However, software solutions do not currently improve efficiency on the scale that hardware improvements do, resulting in a greater need for hardware solutions.

As AI chips become more efficient and have higher performance, they are using extreme amounts of energy. A report from the Internation Energy Agency estimates that data centers and

data transmission networks are responsible for 1% of energy-related greenhouse gas emissions, accounting for around 330 Mt of CO2 emissions in 2020 (“Data Centres & Networks”). While high performance is important in the innovation of AI chips, it is necessary to make the process as sustainable as possible by having a high energy efficiency. Studies show that data centers using TPUs use significantly less power compared to their GPU counterparts. Also, as seen with the ANE, edge AI chips designed for devices like phones reduce reliance on cloud computing, locally processing data, reducing energy consumption. Not only does this achieve low energy consumption, but also low latency, enhancing user experience and sustainability. While current technology is already advancing at an extreme pace, future technologies are becoming increasingly promising. Photonic chips are one of these advancing frontiers in AI chip design. Photonic chips use light instead of electricity to execute operations, allowing them to achieve higher energy efficiency, as light travels much faster and generates less heat than electricity (“What Is a Photonic Integrated Circuit”). Even in its early stages, photonic chips are already improving on silicon ones, demonstrating its potential to change how we think about AI chip design. For AI chips to be made more efficient not only in terms of performance, but in terms of energy consumption, much further research must be conducted, as the limits of chip design continue to be redefined.

This essay has explored multiple techniques from different approaches for optimizing artificial intelligence (AI) chip design, including hardware, software, and algorithmic approaches. The case studies of notable AI chips from leading companies such as Google’s TPUs, Apple’s ANE, and NVIDIA’s GPUs have been discussed and exemplify the dependency of optimization approach on intention. As AI models continue to grow in both complexity and usage, their performance standards and energy standards will only continue to rise. For true optimization of AI chips, both performance and energy efficiency must be considered, as they often come at the cost of the other. Further research may include analyzing the monetary side of AI chip

development and how energy efficiency may come at a higher price or the emerging frontier of quantum computing. For a balance to be found between high performing artificial intelligence and sustainable energy usage, further research is necessary.

Bibliography

Abidi, Yadullah. “What Is CPU Thermal Throttling and How Does It Affect Performance?” *MUO*, 15 Apr. 2023, www.makeuseof.com/what-is-cpu-thermal-throttling/.

“CUDA Cores vs Tensor Cores – Which One Is Right for Machine Learning.” *AceCloud*, 16 Dec. 2024, acecloud.ai/resources/blog/cuda-cores-vs-tensor-cores/.

“Data Centres & Networks.” *IEA*, www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks.

Howard, Andrew G., et al. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.” *arXiv.Org*, 17 Apr. 2017, doi.org/10.48550/arXiv.1704.04861.

Jouppi, Norman P., et al. “In-Datacenter Performance Analysis of a Tensor Processing Unit.” *arXiv.Org*, 16 Apr. 2017, arxiv.org/abs/1704.04760.

Jouppi, Norman P., et al. “TPU V4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings.” *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 17 June 2023, pp. 1–14, https://doi.org/10.1145/3579371.3589350.

Bigelow, Stephen J. “What Is a Tensor Processing Unit (TPU)?” *WhatIs*, *TechTarget*, 16 July 2024, www.techtarget.com/whatis/definition/tensor-processing-unit-TPU.

“NVIDIA A100 Tensor Core GPU Architecture.” *NVIDIA.*

Oh, Fred. “What Is Cuda?” *NVIDIA Blog*, 27 Jan. 2022, blogs.NVIDIA.com/blog/what-is-cuda-2/.

Okemwa, Kevin. “Microsoft Reportedly Acquired the Most NVIDIA GPUs Compared to Its Rivals, Including Google and Meta, for Its AI Projects - Translating to 485,000 Chips and $31 Billion in Expenditure.” *Windows Central*, 18 Dec. 2024, www.windowscentral.com/microsoft/microsoft-reportedly-acquired-the-most-NVIDIA-gpus-compared-to-its-rivals-including-google-and-meta-for-its-ai-projects-translating-to-485-000-chips-and-usd31-billion-in-expenditure.

Sarkar, Arjun. “Understanding EfficientNet‑The Most Powerful CNN Architecture.” *Medium*, 8 May 2021, arjun-sarkar786.medium.com/understanding-efficientnet-the-most-powerful-cnn-architecture-eaeb40386fad.

Swain, Gyana, et al. “Single-Core vs. Multi-Core CPUs.” *Network World*, 14 Jan. 2025, www.networkworld.com/article/971425/single-core-vs-multi-core-cpus.html.

Synopsys. “What Is a Photonic Integrated Circuit (PIC) and How Does It Work?” *Synopsys*, www.synopsys.com/glossary/what-is-a-photonic-integrated-circuit.html.

“Understanding 3D Chip Stacking.” *Assured Systems*, www.assured-systems.com/faq/understanding-3d-chip-stacking/.

Urwin, Matthew. “What Is a Transistor? (Definition, How It Works, Example).” *Built In*, 11 Sept. 2024, builtin.com/hardware/transistor.

“What Is ASIC? - ASIC Cost.” *Arm*, www.arm.com/glossary/asic.

“What Is Edge AI?” *IBM*, 6 Jan. 2025, www.ibm.com/think/topics/edge-ai.

Zadeh, Ali Hadi, et al. “Gobo: Quantizing Attention-Based NLP Models for Low Latency and Energy Efficient Inference.” *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Oct. 2020, pp. 811–824, https://doi.org/10.1109/micro50266.2020.00071.

Zafrir, Ofir, et al. “Q8bert: Quantized 8bit Bert.” *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS)*, Dec. 2019, pp. 36–39, https://doi.org/10.1109/emc2-nips53020.2019.00016.

Zubarev, Vasily. “Computational Photography Part I: What Is Computational Photography?” *DPReview*, 3 June 2020, www.dpreview.com/articles/9828658229/computational-photography-part-i-what-is-computational-photography.