ARM and Machine Learning

Michael Hsiu, Vu Pham, Matthew Thibault

CS533-Computer Architecture, Master of Science in Computer Science, School of Technology and Computing, Seattle City University

[hsiumichael@cityuniversity.edu](mailto:hsiumichael@cityuniversity.edu)

[phamvu@cityuniversity.edu](mailto:phamvu@cityuniversity.edu)

[thibaultmatthew@cityuniversity.edu](mailto:thibaultmatthew@cityuniversity.edu)

**Abstract**

Machine learning is a programming strategy that uses data and algorithms to copy the way in which humans learn. As technological advances have improved storage and processing power for all manner of computational devices, machine learning and AI in general have found their way into a wide variety of industries and applications. Even more recently machine learning has been paired with ARM-based systems, this is notable because until fairly recently storage and processing power limitations for many ARM-based systems prohibited this pairing. This study intends to better understand the developments that have allowed machine learning and ARM to work, explore cutting edge applications of this kind, and identify any industry standards that may exist.

**Keywords:** ARM, machine learning, Neural Processing Unit, embedded AI, edge computing,

Machine Learning Processor, Internet of Things, technology

**1.** **INTRODUCTION**

This project aims to explore several points surrounding machine learning and ARM systems. First, we intend to determine the major constraining factors that have limited the use of machine learning on ARM systems. Second, we will identify hardware and software changes that are being implemented to address these factors and improve the ability of ARM systems to use machine learning. Third, we will look at real world applications that have seen or are predicted to soon see machine learning used in conjunction with ARM systems.

**2. LITERATURE REVIEW**

ARM is an acronym that stands for Advanced RISC Machine, it was originally designed by Acorn Computers in 1985. In 1990 ARM was spun off from Acorn Computers and became an independent company, also called Arm (or Arm Holdings). ARM6 was the first design by Arm as a new company and was the successor of ARM3. It was around this time that Apple began to work with ARM and ultimately used it for processors in Apple’s Newton PDA. While working with ARM, Apple had the opportunity to influence the design changes that were implemented in ARM6. Afterwards Apple would move away from ARM for a time, nonetheless ARM6 became quite popular.

This pattern of collaboration between ARM and other tech companies such as Apple continued over the years. The developers of ARM would work with their licensees to determine processing needs and introduce changes to the chips and instruction sets to meet those needs. Fast forward to today and the leading edge of computation is in machine learning.

Machine learning has a long history dating back to the 1960’s with the search for artificial intelligence. However, AI at the time became dominated by inductive logic programming and machine learning fell out of favor. Then in the 1990’s ML saw a resurgence partly as a consequence of a general shift in goals. Before this point the objective was to create true artificial intelligence but afterwards the goal became to solve real practical problems. This change moved the field in the direction of statistics and probability theory.

Machine learning today relies heavily on statistical analyses to aid in the collection and preparation of data sets used to train ML models as well as test them. In addition, statistical hypothesis tests and estimation statistics are used to configure the models themselves as well as evaluating the models’ results after they are tested and choosing a particular model as the solution to a given problem. (Turing, 2022) It also tends to be a resource intensive task, gathering and preparing large sets of data then creating, training, and testing many different variants of a model to create a single best model for a specific problem is relatively expensive. Additionally, models themselves can be highly scalable allowing for greater complexity in order to achieve greater predictive accuracy, which has kept ML largely in domains with access to large amounts of resources, such as cloud computing. Google’s tensor processing unit (TPU) is a relatively recent innovation that has enhanced the use of ML in cloud computing by designing a processing unit that is optimized for machine learning algorithms while removing hardware and instruction support for other kinds of tasks.

Today Arm CPUs are used in about 90% of mobile applications, including smartphones, tablets, and more as well as having a significant presence in the GPU market. Arm processors are also used in about 90% of IoT (Internet of Things) applications such as smart watches, smart TVs, home environment control, fitness machines and trackers, etc. (Alsop, 2022) In addition, Arm intends to increase its market share in networking equipment, cloud computing, and vehicle infotainment. Due to its significant presence in a variety of technology markets and the increasing demand for machine learning across all sectors, Arm is in a similar position to Google, to be a leader in bringing ML to a wider range of technologies.

**3. CONSTRAINTS**

ARM architectures are largely designed for, and prevalent in mobile phones and embedded systems due to the low equipment cost and simple design. This is a unique constraint for this architecture when it comes to deploying AI models and using machine learning on these systems. First of all, the computational power of ARM devices is less than traditional desktop CPUs and especially less than GPUs. The capabilities of computational power are usually used for training and sometimes for inferencing complex machine-learning models. Additionally, ARM processors are designed to consume less battery because of their RISC architecture. This factor is perfect for mobile phones and other mobile systems since users don’t really need to charge the devices while they are using them. However, the design can affect the use of machine learning on the system because training or running complex machine learning models can be power-intensive, leading the devices to battery drain and thermal throttling, which can decrease the performance of the task.

Storage is another bottleneck in ARM systems for machine learning. The ARM-based system is mostly for phone and tablet or any IoTs segment. These devices usually have less RAM compared to desktop and server setups. Loading large models or datasets from storage can be time consuming. This can affect the latency in machine learning tasks. Besides, using ARM system can limit the number of models that you can use for machine learning.

Machine learning in general benefits from a high degree of scalability. In essence, larger amounts of data, and longer and stricter training regimens generally allow for more complex and accurate AI. Due to this ML is often performed in resource rich environments like cloud computing to create the best possible models. However, training and inferencing from such complicated models is simply not practical for many real-world problems that could benefit from a tailored ML solution. A variety of tasks from managing vehicle congestion in real time to automatic home environment control are interesting ideas but difficult to implement in practice. So, there is a very real need for cheaper, more efficient ML solutions that are still accurate enough and fast enough for the tasks they are intended to resolve.

**4. NEURAL PROCESSING UNIT AND ARM NN**

Due to the tight constraints faced by many ARM-based systems, there was a need to develop new hardware that could improve their compatibility with machine learning. Many machine learning algorithms have workloads that are easily broken up and run in parallel which makes these algorithms a poor match for CPUs. In general, GPUs are much more efficient hardware for performing this kind of work, but they are also more complex than they need to be. So, it seemed that there was niche that could be filled by a new kind of circuit.

A neural processing unit (NPU) is similar to a GPU in its ability to handle many parallel tasks at once while also having significantly simpler logic since the kinds of tasks they handle are highly regular. The simpler logic saves space and allows these circuits to have more multiply-accumulate (MAC) units which is critical because deep neural networks can require billions or even trillions of MAC operations. NPUs are grouped into 2 categories: training and inference. As one would expect, training NPUs are designed to accelerate the training of new machine learning models. On the other hand, inference NPUs are designed to use existing models to process input data (such as images or voice clips) and categorize them.

**Machine Learning Processor**

Introducing NPUs to heavily constrained systems, such as mobile phone system-on-a-chip (SoC), has allowed these devices to make use of machine learning more efficiently while freeing up the CPU and GPU for other tasks. Two such series of processing units are Arm’s Ethos-N series and their Ethos-U series, both of which utilize Arm’s first commercial neural processor microarchitecture dubbed simply, Machine Learning Processor (MLP). The MLP architecture was designed to be a significant improvement over the company’s existing Cortex CPUs and Mali GPUs with an emphasis on CNNs (convolutional neural network) and RNNs (recurrent neural network). It was also created to be highly scalable so that the architecture could operate effectively in very low power IoT applications and also as a part of complex mobile or network SoCs. (WikiChip, n.d. b)

A diagram of a computer

Description automatically generated

**Figure 1:** Arm’s Machine Learning Processor Architecture.

*Note*. From *AnandTech* [Image], by Frumusanu, 2018, AnandTech. <https://images.anandtech.com/doci/12791/2.PNG>

The scalability of the MLP architecture comes from the variable amount of compute engines that a particular chip comes with. Compute engines are grouped into quads and a chip may have a single quad (with a single compute engine instead of the maximum of 4) or many. Each compute engine has 3 main parts: the SRAM, the MAC compute engine (MCE), and the programmable layer engine (PLE).

Arm’s MLP architecture has been implemented in two series, Ethos-N and Ethos-U. The Ethos-N series are fully independent NPU’s that can be built into any SoC application and several of these can also be combined for higher performance. The latest in this series is the Ethos-N78. The Ethos-U series is a microNPU, meaning they do not fully implement the MLP architecture. Instead, they are designed to work together with an Arm Cortex-M processor instead of a PLE and typically have a single compute engine and no dedicated SRAM. This series is intended for use in deeply embedded ML applications, the latest of which is the Ethos-U65. (WikiChip, n.d. a)

**ARM NN**

With the introduction of NPUs, hardware that better supported machine learning, Arm designers were also challenged with making that hardware usable by their customers. Arm NN is a software developement kit that allows the use of existing neural network frameworks on Arm products including Cortex-A CPUs, Mali GPUs, and Ethos NPUs. When used on Cortex-A and Mali chips the framework makes use of Arm’s compute library to have access to operations that make the most use these chips. This SDK is designed to translate common ML frameworks such as TensorFlow, Caffe, and others to Arm’s NN framework, reducing the entry requirement to work with ML on Arm products. Additionally, Arm NN works with NNAPI, Google’s interface for accelerating neural networks on android devices. (Arm Ltd., n.d. a)

**5. APPLICATIONS**

Much of the ARM machine learning has been implemented in three main ways. The following is quoted from the Arm Ltd. (n.d. b):

- Vision: Deliver immersive visuals and

capture insights from intelligent cameras.

Examples include image classification,

object detection, image segmentation,

super resolution, human pose estimation,

face recognition, and depth estimation.

- Voice: Enable key word detection and

automated speech recognition locally on

the device – with no cloud required.

Examples include key word spotting,

automatic speech recognition, natural

language processing, beamforming, noise

suppression, machine translation, and

speech synthesis.

- Vibration: Leverage vibration to analyze

signals, monitor health, predict

maintenance, and detect anomalies.

Examples include human activity

recognition, cardiac abnormality detection,

industrial anomaly detection, sensor

fusion, motor control, and predictive failure.

Notice that many of these applications have use cases that are small in scale, remote in location, or numerous in quantity and therefore may not benefit from being connected to a network with access to cloud computing resources. This is the current target market for Arm ML, lightweight ML solutions being made available in circumstances where cloud computing is unavailable or too costly to implement.

As mentioned earlier, the compute library from the Arm NN framework enables the efficient use of ML on existing legacy systems or in very lightweight situations where a dedicated NPU may not make sense. Furthermore, convolutional neural networks are a class of neural network that Arm’s MLP architecture is particularly effective at operating. A CNN is a type of deep learning algorithm that is great for image recognition, making Arm NPUs an excellent match for visual tasks like facial recognition and object detection as well as many others.

**Case Study: NetsPresso**

NotaAI is a South Korea based AI startup that uses their proprietary Automatic AI Model Compression Platform, NetsPresso, to bring AI to tiny applications. The platform specializes in quickly and efficiently compressing complex deep learning AI models for use on individual devices without compromising performance. (Arm Ltd., n.d. c)

NotaAI has used their software to create ML solutions to many image recognition problems, largely in the domain of traffic safety. Several of their successful ventures include pothole detection on vehicle dashcams, fire detection for CCTVs, and an Infrared pedestrian detection system for traffic safety agencies. (NotaAI, n.d.)

All of these solutions were developed for use on the Jetson Nano which is a standalone board and competitor to Raspberry Pi. It is powered by a Quad-core ARM Cortex-A57 MPCore processor. (Nvidia, n.d.)

Additionally, the NetsPresso platform has been successfully utilized to upgrade existing home security solutions to use facial recognition where previously only fingerprint and keycode solutions had been implemented. The developers were able to do so by compressing their ML model to fit onto the legacy system with no hardware upgrades required.

**Case Study: Ignitarium**

Ignitarium is a product engineering design company from India. It’s mainly known for multimedia, DSP and deep learning. They perform research focused on combining DSP and deep learning to have software that differentiates video and audio-based products.

The company had the goal of dealing with background noise from daily life which can adversely affect work performance and convenience. Ignitarium has implemented a small memory footprint, a deep learning-based real-time noise suppression that can run on a low-cost micro-controller. They figured out that the traditional SDP is not sufficient in this case. It requires certain hybrid models including a deep learning-based approach. Deep learning can help with training models in a variety of conditions.

There are some main features that Ignitarium’s Real-time Noise Suppression (IGN-RNS) can bring to customers: deep learning base, small memory footprint, low latency, different sampling rates are supported and optimized for high-performance.

The main challenge while developing the algorithm is processing memory and power. With the RAM of most devices being under 30 kb, Ignitarium chose a suitable NN architecture and implemented it to suit the limited resources in processing memory and power. Cortex-M is the most reasonable computing platform choice. Ignitarium implemented Arm Cortex-M4 for their solutions. Arm has several advantages such as bringing a strong ecosystem, power efficiency and mature software support for deep learning.

**6. CONCLUSION**

The constraints of edge computing are at odds with resource intensive AI solutions. To address this discrepancy in resource availability, ARM, a leader in the IoT and edge computing technology has been working with stakeholders to develop new tools, like their ARM NN SDK, and hardware, like their Ethos series of neural processing units, to bring greater efficiency and performance to ML algorithms. These improvements also enhance the interoperability of the ARM architectures with existing ML models thereby reducing the entry requirements to apply ML in this previously out of reach domain.

In the domains of embedded AI and edge computing the current industry practice appears to be to develop ML models on higher powered machines, then compress them and embed them into the more constrained environments where they will actually perform. This process is tricky and prone to error but platforms like NetsPresso are on the leading edge to automate the process of bridging the gap from the training environment to the functioning environment and allow faster and cheaper production of unique ML solutions. In the future, even closer integration of training and functional environments seems likely, reducing the tension between the competing interests of compression (to fit models on embedded systems) and the prediction accuracy and speed of AI models.

**7.** **REFERENCES**

Arm Ltd. (n.d. a) *Arm NN SDK.*

<https://www.arm.com/products/silicon-ip-cpu/ethos/arm-nn>

Arm Ltd. (n.d. b) *ML Solutions - Artificial intelligence.*

<https://www.arm.com/markets/artificial-intelligence/machine-learning>

Arm Ltd. (n.d. c) *Partner Solution Brief – Nota*

*Inc – NetsPresso.* <https://armkeil.blob.core.windows.net/developer/Files/pdf/ai-ecosystem-catalogue/nota-ai-ecosystem-catalogue-solution-brief-v1.pdf>

Arm Ltd. (2020) *Arm Ethos-N78 Processor.*

<https://armkeil.blob.core.windows.net/developer/Files/pdf/ML%20on%20Arm/arm-ethos-n78-product-brief.pdf>

Alsop, T. (2022, April 8) *Arm’s market share and targets across key technology markets in 2019 and 2028 fiscal years.* Statista <https://www.statista.com/statistics/1132112/arm-market-share-targets/>

Frumusanu, A. (2018, May 22) *ARM Details “Project Trillium” Machine Learning Processor Architecture.* AnandTech.

<https://www.anandtech.com/show/12791/arm-details-project-trillium-mlp-architecture>

NotaAI. (n.d.) *NetsPresso Case Study.* <https://www.nota.ai/netspresso>

Nvidia. (n.d.) *Jetson Nano.* <https://developer.nvidia.com/embedded/jetson-nano>

Takahashi, D. (2019, October 20). *How arm wants to bring machine learning to ordinary computing devices*. VentureBeat. <https://venturebeat.com/ai/how-arm-wants-to-bring-machine-learning-to-ordinary-computing-devices/>

Turing. (2022, February 11) *Introduction to*

*Statistics for Machine Learning.*

<https://www.turing.com/kb/introduction-to-statistics-for-machine-learning>

Utmel Electronics. (2021, December 28) *Neural Processing Unit (NPU) Explained.* <https://www.utmel.com/blog/categories/integrated%20circuit/neural-processing-unit-npu-explained>

WikiChip (n.d. a) *Ethos – Arm.* Retrieved August

29, 2023, from <https://en.wikichip.org/wiki/arm_holdings/ethos>

Wikichip (n.d. b) *Machine Learning Processor*

*(MLP) – Microarchitectures – ARM.* Retrieved August 29, 2023, from <https://en.wikichip.org/wiki/arm_holdings/microarchitectures/mlp#Programmable_Layer_Engine_.28PLE.29>

**8. WORKLOAD ASSIGNMENT**

Michael Hsiu: Applications, Powerpoint

Vu Pham: Applications, Powerpoint Presentation, Constraints

Matthew Thibault: Abstract, Introduction, Literature Review, Constraints, Neural Processing Unit and ARM NN, Applications, Conclusion, References