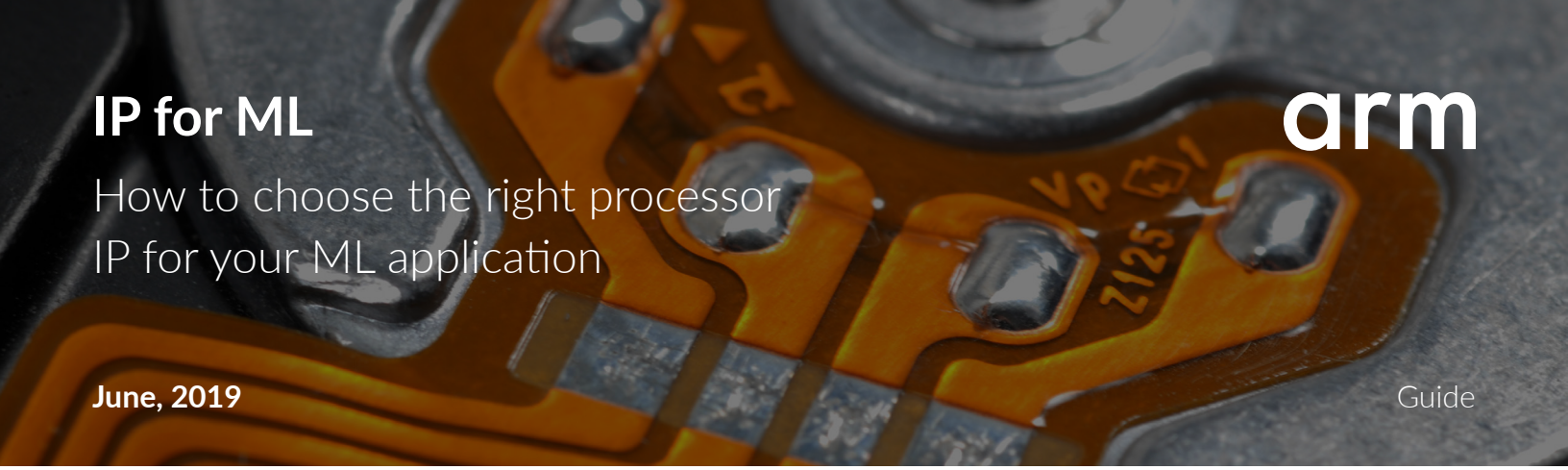


IP for ML

How to choose the right processor IP for your ML application

June, 2019

Guide

Machine learning (ML) is becoming increasingly powerful at solving tough computational tasks that drive a myriad of AI applications. ML has already proved its value in an array of use cases—everything from facial recognition for unlocking smartphones and real-time voice and video stream analysis, to Big Data analytics and even the monitoring of vibrations in industrial machinery.

The value of ML increases as it moves out of the cloud and the data center, and into a growing variety of edge devices and applications—from self-driving automobiles and internet of things (IoT) environments to consumer mobile devices and automated services. And as these trends create more connected devices that require ML capabilities, the universe of potential ML use cases is poised to grow larger, too.

But this growth also brings a new set of challenges, as developers adapt ML applications to a much wider range of use cases and performance requirements than ever before. For manufacturers of ML-enabled IoT devices to fulfill their vision of intelligent products everywhere—whether a wearable fitness tracker, smart speaker, or industrial machinery sensor—they must make critical decisions regarding the hardware and software used in ML system design.

Already the default processor for AI computing is the CPU, whether it's handling the AI entirely or partnering with a co-processor, such as a GPU or an NPU for certain tasks. Arguably, the CPU will remain the workhorse for ML workloads because it benefits from common software and programming frameworks and serves as mission control in more complex systems that leverage accelerators. This guide explores key considerations to take into account when choosing the right mix of processor IP for an ML application to achieve an optimal balance of ML system performance, cost, and product design. It also offers advice on how to approach these critical decisions.

How Does ML Work?

ML performs computational tasks by recognizing patterns and making inferences, rather than relying on explicit instructions. ML algorithms build models using sample data—a process referred to as "training"—and then applies those models to accomplish a defined task. This process is referred to as inference. ML algorithms are capable of learning and improving over time, delivering more accurate results and adapting to changes. This makes ML ideal for computational tasks that are impossible, or at least cost prohibitive, using hand-coded algorithms.

Balance Performance, Power, Cost, and Design

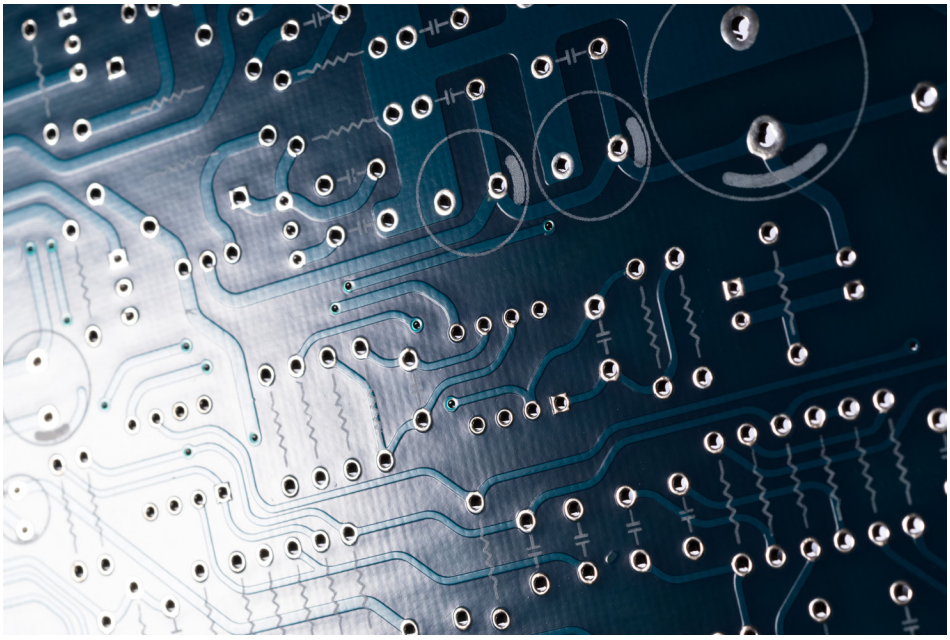
As ML applications become more varied and more complex, it's critical that you choose the right processor IP for your use case. Your choice of IP should take into account several factors and will ultimately come down to a balance of performance, power, cost and design. Several key considerations will help you strike that balance and choose the best-performing and most cost-effective IP.

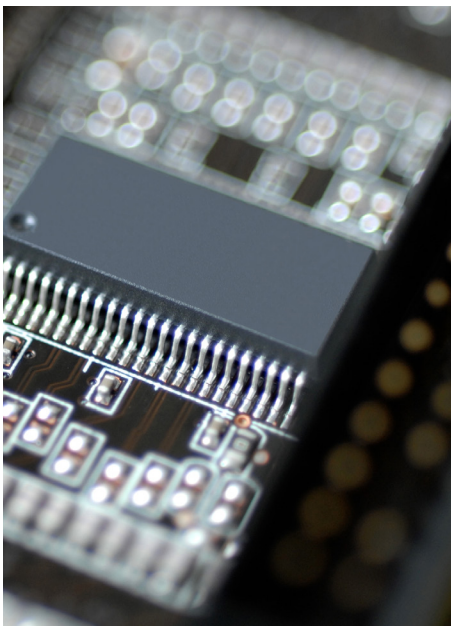
Understand Your Use Case

Defining your machine learning use case and workload may be the single most important factor in choosing the right processor IP. The questions you ask to understand your use case may vary, but the goal is always the same: to set realistic priorities, identify constraints, and define critical requirements for your ML system.

Questions to Ask to Define Your ML Use Case

1. **What are the goals of your ML application?**
2. **Is performance a primary concern?**
   Performance, and specifically response times, are a critical concern for creating high-quality user experiences. In other cases, such as long-term analytical tasks, performance may be a relatively unimportant concern.
3. **Will the ML application handle a high-intensity workload?**
   Object recognition within a video stream, within a facial-recognition app, under vibration, and within a still-image archive involve similar ML tasks—but very different workload volumes.
4. **Where and how will the ML application satisfy its power requirements?**
   Implementing ML within a mobile app, for example, makes power consumption a far more important concern than it would be within a data center environment.





Maintain a Flexible Approach

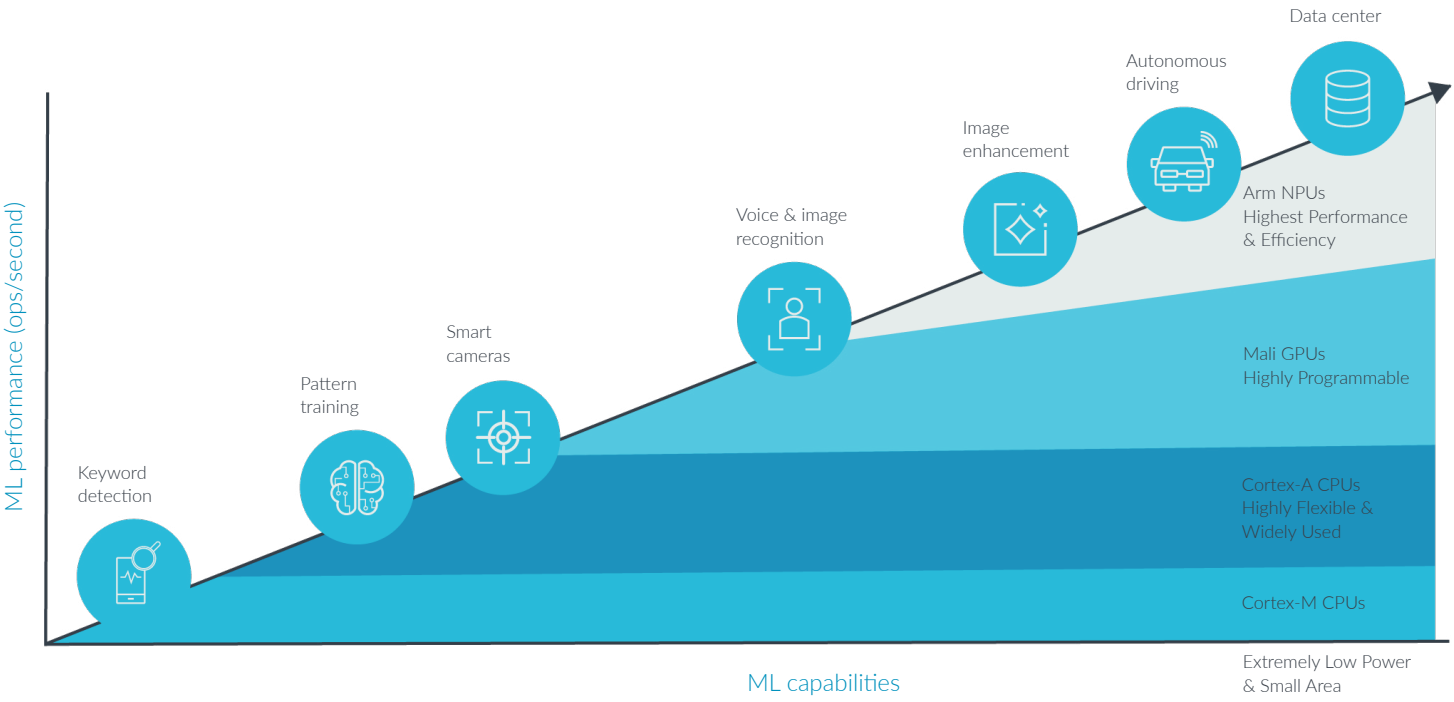
Power, performance and area (PPA) represent one of the stark realities of semiconductor design and gaining an advantage in one area often means trading off capabilities in others. But tradeoffs are rarely simple or straightforward. When evaluating different IP, be sure to think about what you should prioritize and what you can sacrifice when it comes to performance, cost, and power.

Power: Will the device be 'always-on'? What is your power target? What are your power constraints? Answers to these questions may lead you to prioritize power.

Performance: What is your performance bar? Is this likely to change given market or competitive conditions?

Area: Consider whether an existing processor has spare capacity and determine your area budget for new ML IP. What is the market forecast for both demand and potential competition?

A critical lesson here is that the decisions you make today to optimize PPA are likely to change tomorrow. An ML platform designed for flexibility—allowing you to shift between IP options at different points on a PPA spectrum—ensures that your machine learning strategy can adapt and evolve as needed.



Typical ML Hardware Choice

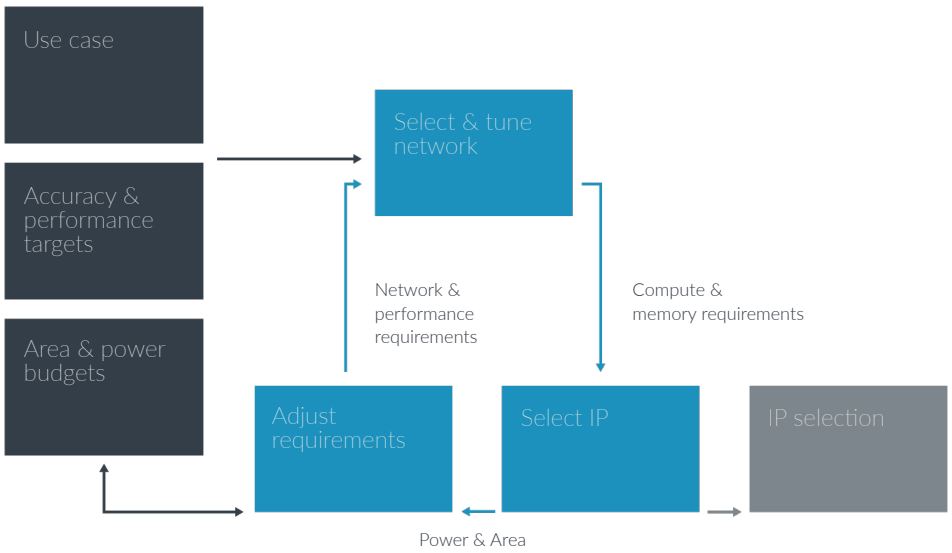
"Best practice: Look for machine learning IP that addresses the hidden risks of data-flow and memory bandwidth bottlenecks."

Understand the Role of Neural Network Models

Neural networks are the algorithmic foundation for many modern ML applications, and it's important to understand the three key traits that define today's neural network models to choose the best IP for a machine learning platform:

1. **The neural network model landscape is crowded and very diverse.**
   In this environment, it is useful to make IP choices that work efficiently with the widest possible range of models.
2. **Neural network hardware preferences vary widely.**
   Some algorithms are more effective working with CPU or GPU architectures; others are better suited for specialized ML processors. Many organizations find it necessary to work with algorithms that span both approaches.
3. **Neural network models continue to evolve rapidly along multiple paths.**
   An effective ML platform and IP choice must be able to adapt and evolve to keep pace with these changes—for example, through the use of programmable layer engines, rather than fixed-function blocks.

After you choose a neural network it's necessary to train it so that it can eventually perform as intended. To train your network, make sure training data is available and of sufficient quality.



The IP Selection Cycle

Determine Required System Resources

Make sure you determine the resources your ML application will require:

- What is the required compute capacity?
- How much SRAM and DRAM do you need?
- How much bandwidth to external memory is required?

Look for IP with features such as lossless feature map compression, and system designs that maximize the use of available memory bandwidth—and think carefully about whether your use case presents any special challenges around data-intensive or high-performance workloads.

Futureproof Your Investment

While you can't necessarily predict now how your ML application will evolve, the ML IP and platform you choose must be able to support your growth in the market. Therefore, be sure to invest in IP with a proven track record of leading new technologies and supporting innovations. This IP must be able to:

- Scale to accommodate a wide range of hardware and use cases
- Adapt and evolve quickly and efficiently
- Offer a full spectrum of options for performance, power, cost, and design

These are the qualities that will set apart successful machine learning strategies, no matter how the technology evolves.

Industry-Leading IP from Arm

As the growth of intelligent IoT devices continues and AI-based systems continue to supplant web services and apps, Arm IP designs continue to play a critical role. Arm is bringing AI to trillions of edge devices, adding ML capabilities to processor technology to make them smarter, more energy efficient, and more affordable.

Arm's Project Trillium is driving the AI revolution, redefining device capabilities through a thriving design and development ecosystem that has extended its reach from smartphone CPU design and app development to ML.

Project Trillium provides flexible support for ML workloads across all programmable Arm IP, as well as partner IP with a new class of ultra-efficient processors and highly optimized software, including:

- The Arm Machine Learning processor with fixed-function and programmable engines, optimized for performance and energy efficiency.
- An open-source developer kit, Arm NN, that allows seamless integration with existing neural network frameworks, such as TensorFlow, Caffe, and Android NN.

With support from a vibrant and diverse ecosystem, Project Trillium is driving innovation and choice.

For more information on Arm processor IP and software, visit the AI Solutions section of our website, or contact us.

arm