**Artificial Intelligence Where You Need It.**

**From Cloud to Edge to IoT Endpoint**

<https://www.arm.com/resources/report/scalable-machine-learning>

November 11, 2019   —  [Arm Enables](https://www.arm.com/blogs/blueprint/arm-enables)

# Scalable AI Platforms Put Compute Where It Counts

## **We need to design systems capable of dynamically adjusting the type—not just the speed—of processing resource they can deliver**

**By**

[**Steve Roddy, VP Machine Learning Group, Arm**](https://www.arm.com/blogs/blueprint/author/steve-roddy)



Wear and tear on big, heavy vehicles such as trains can cause unexpected delays and repairs, not to mention create safety hazards that can go unnoticed for months until they become critical. In the past, maintenance teams personally examined the undercarriage of a locomotive to look for stress cracks and other anomalies. Later, imaging and sonar technologies were introduced to find what the human eye could not.

But today, transportation companies are leveraging new digital technologies that monitor the condition of vehicles within their fleet at the points of concern and can alert drivers and maintenance teams before problems can arise. Key to this is [artificial intelligence](https://www.arm.com/solutions/artificial-intelligence?utm_source=blueprint&utm_medium=refferal&utm_campaign=blueprint) (AI), specifically its [machine learning](https://www.arm.com/solutions/artificial-intelligence/machine-learning) ([ML](https://www.arm.com/solutions/artificial-intelligence/machine-learning)) subset.

**“Make any attempt to impose restrictions on the way an ML model is built and you’re likely to irreparably compromise its accuracy.”**

German sensors’ company Lenord+Bauer is an industry leader in AI-based solutions that rail companies use to keep their equipment safe and in optimal working condition. But the key is that Lenord+Bauer isn’t leveraging the cloud to analyze all the vibration, heat and speed data its sensors capture as the trains chug along. It’s running neural networks directly on an Arm Cortex-M based ST Microelectronics STM32 microcontroller and transmitting the results locally.

By putting its edge [AI solutions](https://www.arm.com/solutions) [directly at the sensor level](https://www.youtube.com/watch?v=iHkEOHqmdb8&feature=youtu.be), Lenord+Bauer realizes lower power consumption and low latency. The overall solution gets a lift from [STM32Cube.AI](https://blog.st.com/stm32cubeai-neural-networks/), launched at 2019 CES, an advanced toolkit capable of interoperating with popular deep learning libraries to convert any artificial neural network for STM32 microcontrollers to run optimized inferences.

### **Cloud, edge, endpoint**

This type of solution is a prime example of scalable AI platforms today, which have taken advantage of continuous improvements and innovations in processor technologies from the cloud all the way to edge and endpoint systems. If the early days of AI and ML were about sending data to big cloud data centers to be analyzed and acted upon, that type of compute is today being distributed out via the edge to endpoint devices, cutting cost, latency and increasing security and improving customers’ experience.

But while this approach to distributed computing seems a natural evolution, it’s in fact changing decades of design methodology in embedded systems.

Embedded systems are developed within performance parameters, an envelope based on power, cost, heat dissipation, size, weight and any number of measurables that can be traded off against each other to meet defined targets. Historically, the role of an embedded developer was to write a piece of safe, predictable code capable of performing its task within the limitations of those performance parameters. Thus, the idea that code’s characteristics might change after it has been deployed remains, to many embedded engineers, stuff of nightmares.

### **Scalable AI platforms are the new norm**

Yet this is the very nature of ML; accuracy is often twinned with hardware capability. Make any attempt to impose restrictions on the way an ML model is built in order to comply with constrained parameters and you’re likely to irreparably compromise its accuracy. In the case of mission-critical monitoring of locomotive wear, that’s not an acceptable risk.

Scalable AI platforms provide the solution, though this isn’t the same as frequency scaling, where the frequency of a microprocessor is automatically adjusted to consume less power or provide more processing capability. This technology was great news for embedded systems when it was first developed well over a decade ago, but it was essentially just a sliding scale that allowed a fixed architecture to run hotter or cooler depending on the job at hand. This kind of scaling isn’t going to be enough to deliver the power and performance needed in the embedded devices now being developed to run ML models.

Instead, we need to design systems capable of dynamically adjusting the type of processing resource they deliver based on a given task – changing the effort rather than simply increasing or decreasing it. The reason for this is simple: The path to inference is littered with variables, and with so many layers of probability to work through, any one of those variables stands to change the path completely.

Take natural language processing and speech recognition as an example: the speaker’s voice and cadence will all play a role in the efficacy of the model, yet there may also be interplay between these parameters that result in a different experience under various conditions. Simply increasing the clock frequency to meet the inference target isn’t guaranteed and will likely bust the power budget, without improving accuracy.

### **Neural Processing Units (NPUs) tackle AI heavy lifting**

While current [CPU](https://www.arm.com/why-arm/architecture/cpu) architectures can and are being used for ML, today’s architectures almost certainly can’t provide the most optimal way of executing them. Yes, models can run on CPUs using all the usual ALU features found in most processors. They can also benefit from highly parallel architectures that feature massively multiple instances of these features, such as GPUs. But it’s already clear that GPUs are not the best way we will ever come up with for executing ML models.

In fact, we already have examples of neural processing units (NPUs), and the semiconductor industry is hard at work developing entirely new architectures for executing ML models more efficiently. These ‘chicken and egg’ scenarios rarely end with the most optimum solution, at some point either the hardware or the software becomes fixed in order to let the other move forward. The right way to address this is to commit to a common software framework that can be used across compatible but scalable hardware platforms, so that both evolve together in lockstep.

### **Flexible heterogeneous architectures**

By doing this, the scalable AI platforms needed to support intelligent compute can be extended from the core of the network to the very edge, without locking the architecture down to a fixed platform. Project Trillium is Arm’s all-inclusive heterogeneous ML compute platform comprising cores and software. Originally developed to answer the hardware requirements needed by mission-critical endpoint devices such as those used in locomotive maintenance or healthcare, Project Trillium is now expanding to address ML at every point in the network.

The common software platform here is Arm’s neural network software libraries, Arm NN, that can run across [Arm processor](https://www.arm.com/products/) platforms and are also compatible with leading third-party neural network frameworks. The hardware includes the existing [Arm Cortex-A](https://www.arm.com/products/silicon-ip-cpu) and [Arm Mali](https://www.arm.com/products/silicon-ip-multimedia) [GPU](https://www.arm.com/products/silicon-ip-multimedia) processors that are being enhanced for AI and ML, as well as totally new [Arm Ethos](https://developer.arm.com/ip-products/processors/machine-learning/arm-ethos-n) processors for ML acceleration.

### **Compute where it counts**

In terms of scalable AI platforms, ML can and does run on processors as small and resource-constrained at the Cortex-M class, and as feature-rich as the Mali GPUs. However, truly scalable AI platforms are needed to meet all the needs of ML from cloud, to edge, to endpoint device, which is where the next step in processor evolution comes in. Neural processing units such as Arm [Ethos](https://developer.arm.com/ip-products/processors/machine-learning/arm-ethos-n) represent the new generation of processor architecture that will support ML in more applications. Only Arm offers this range of scalable AI platforms across the ML landscape.

It is very rare for engineering teams to have full access to all of the requirements they will encounter as a project progresses. When ML becomes part of that mix, there is even less opportunity to fix features in stone. It is hard enough to build a platform that will meet the unknown but predictable demands that may come along a year after it is introduced. With ML, the predictable nature of the application is lost. Choosing scalable architectures that can be composed of MCUs, CPUs, GPUs and NPUs will help future-proof hardware platforms against new software applications that haven’t even been conceived yet.

There are many unknowns at play here, in terms of what ML models we’ll be creating in the future, how much compute power they will need to deliver the desired accuracy, how quickly computer scientists will be able to improve models so they need less power—all of these considerations have a direct impact on the underlying hardware.

The only thing we do know is that end users will have expectations that need to be met, and meeting those constantly changing expectations will require a flexible and scalable platform.

**Discover more about how**[**the Project Trillium heterogenous machine learning framework**](https://www.arm.com/products/silicon-ip-cpu/machine-learning/project-trillium)**combines Cortex CPUs, Mali GPUs, and Ethos NPUs to deliver advanced machine learning (ML) use cases.**

WRITTEN BY

[](https://www.arm.com/blogs/blueprint/author/steve-roddy)

#### [**Steve Roddy, VP Machine Learning Group, Arm**](https://www.arm.com/blogs/blueprint/author/steve-roddy)

Steve began his career as an engineer in the 1980s, designing supercomputer processors. With over 20 years’ experience in the semiconductor IP business, he holds patents in cell library architecture and analog design.

**A course at UDEMY:**

<https://www.udemy.com/course/artificial-intelligence-and-machine-learning-fundamentals/?utm_source=adwords&utm_medium=udemyads&utm_campaign=LongTail_la.EN_cc.ROW&utm_content=deal4584&utm_term=_._ag_77879424134_._ad_437497333833_._kw__._de_c_._dm__._pl__._ti_dsa-1007766171312_._li_1001720_._pd__._&matchtype=b&gclid=CjwKCAjwydP5BRBREiwA-qrCGliUIB6o3J3jJTLajmPt-MLLZO9-4w770OFxhq_MVo1EkpsWlIj7GxoCRDAQAvD_BwE>

# Online course at M.I.T.: ARTIFICIAL INTELLIGENCE: IMPLICATIONS FOR BUSINESS STRATEGY

<https://executive-education-online.mit.edu/presentations/lp/mit-artificial-intelligence-online-short-course/?ef_id=c:342482157814_d:c_n:g_ti:kwd-297351945584_p:_k:%2Bartificial%20%2Bintelligence_m:b_a:63771932155&gclid=CjwKCAjwydP5BRBREiwA-qrCGqLma5XcdyROAKJPILIftgJN6ailIrYpCAksvJcRGwZyB8HrksIFehoCkbEQAvD_BwE&gclsrc=aw.ds>