The Emergence of AIoT

Originally published at https://www.iflexion.com/blog/aiot

Published: January 6, 2021 | By Martin Anderson

In the last few years, momentum has grown around the idea of 'local' AI. This kind of on-device machine learning is currently best-known in terms of the recent generation of mobile AI assistants capable of learning to understand and anticipate users' needs over time.

For instance, widely-sold consumer implementations such as Apple's Core ML equip users' devices with stripped-down, templated machine learning models that can adapt and interpret the owner's text input with increasing efficiency over time, using on-board neural network hardware that's designed not to overload the device or use too much power.

Subsequently, business markets have come to expect that the boom in machine learning must or should merge with the evolution of the internet of things (IoT), in order to create a new generation of devices and architectures that can not only gather data, but also process it through local neural networks and other machine learning techniques. This paradigm has become known as AIoT.

There are a few problems with this idea: the central tenets of IoT infrastructure were that it would be simple, low-powered, and cheap to the point of disposability[1]. Conversely, useful machine learning implementations are generally complex, energy-guzzling, and still far from cheap to create or deploy. In fact, machine learning currently represents the most resource-intensiveuse of computing power[2].

In this article, we'll take a look at how this new conjunction of technologies might overcome the hard physics and economics that make it difficult to train and/or run machine learning models without central cloud infrastructure, enabling a new generation of genuinely 'smart' devices and remote data-analysis systems to aid AI software development, consumer needs, and research.

The Economics of IoT Devices

Over the last five decades, Moore's Law correctly predicted that every 18 months computer processing power would approximately double for end users. Unfortunately, there's now general agreement[3] that this particular facet of the tech revolution is exhausted. Most new improvements in this respect involve increasing power levels, which is particularly obstructive for the development of 'local' and mobile machine learning[4].

There remain physical reasons why small devices can't easily become more powerful and useful than they are at the current state-of-the-art without recourse to cloud services (which come with the latency and security issues of a centralized, network-based approach).

All possible solutions involve some kind of sacrifice:

- Lowering power consumption affects availability, versatility, and responsiveness.
- Thinner semiconductors (which might resolve many problems) are expensive and set to hit a hard limit that only new techniques and approaches can solve[5].
- The network connectivity that makes small devices useful is expensive in terms of power drain.
- Security measures such as data encryption (often necessary, sometimes mandated[6]) place additional drain on the device's power consumption.
- Newer, more energy-efficient communication protocols often depend on mesh networks or proximity to better power sources[7] — a useless approach in remote locales.
- The move towards multithreading as an energy-saving measure[8] is not particularly helpful for single-purpose, dedicated devices such as smart monitors — and, in fact, most IoT sensor deployments.

Anyone over 30 understands these restrictions instinctively, remembering the many incremental sacrifices we made as we gradually transitioned from the more powerful desktop-based computing of the 1990s to the less powerful — but more available — mobile devices of the smartphone revolution more than a decade ago.

There are physical reasons why small devices can't become more powerful than they are at the current state-of-the-art without recourse to cloud services.

Industry's Need for Intelligent IoT Infrastructure

Over the last ten years, global interest has grown in developing decentralized, semi-centralized and federated monitoring and regulatory systems in areas and sectors where even basic network infrastructure is poor or non-existent.

Besides the 47% of the world that has no internet access at all[9], there are a number of physically isolated sectors that have always had to depend on expensive dedicated and static technologies — equipment which was shipped to remote locations and required to operate independently *in situ*; without updates, without communication with wider business operations networks, and without the commercial advantages of high capacity data analysis resources.

These isolated sectors have included agriculture and deep sea and shipping operations, as well as architectural integrity monitoring projects and military placements — all of which are either frequently operating with limited or zero network coverage, or else have security issues that preclude the constant use of cloud-based IoT infrastructure.

In these cases, it would be useful if the local sensor assemblies could actually do some of the hard work in processing the data through machine learning systems — especially since the data stream of each one is usually very specific and not always suited for generically trained AI models, as we shall see.

The Logistics of IoT Frameworks

'Dumb' IoT: Shunting Raw Data to the Cloud

An IoT model typically features a sensor transmitting data over the internet to a central processing infrastructure elsewhere on the internet. In most cases, the sensor apparatus itself has a limited or zero processing capacity, being a 'dumb' data capture device capable of sending data to the network and encrypting it if necessary.

Since a primary appeal of IoT infrastructure is the low cost and easy mobility of such sensors, the sensor apparatus may also have little or no storage capacity and lack a hard-wired mains power source. In most cases, data will also be lost for the duration of any local network outages.

If the sensor must communicate data to the cloud over 3, 4 or 5G, it will experience a rising scale of energy consumption[10]; if the information is encrypted, higher still[11]. This means period manual battery replacement, or else the use of more innovative renewable power sources such as solar[12], wind[13], ambient vibrations[14], or even radio waves[15].

This IoT configuration is most likely where the sensor is isolated, such as a traffic-monitoring computer vision camera in a long and remote stretch of road[16], or an installation to monitor for cracks in a building or bridge[17].

Fog-based IoT: 'Near-Device' Data Analysis

In 2011, Cisco coined the term 'fog computing' to indicate a more localized data-gathering/analysis framework[18].

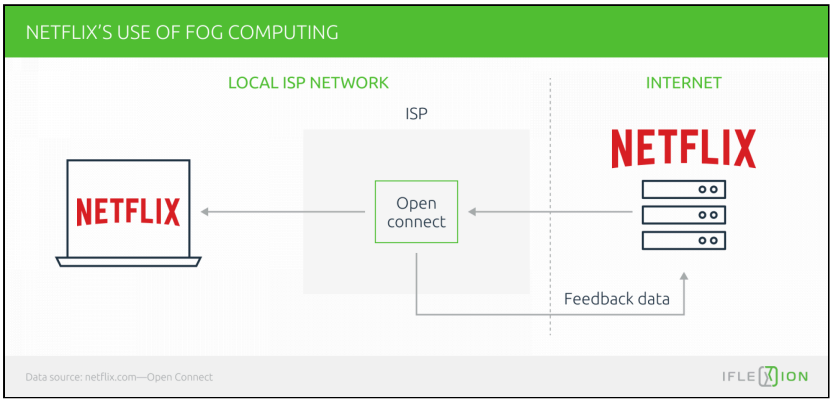
A fog deployment is effectively the same as an IoT deployment (see above), except that the sensor transmits data to a local hub rather than over the internet to a central server. The hub itself may transmit processed or gathered data periodically back to the cloud and will have more power and computing resources than the sensor equipment.

Fog computing is most suited to a large group of low-power sensors that contribute their data input to a more resilient and capable local machine; an example would be an array of agricultural sensors dispersed over a harvest field, with a nearby hub collating and curating the various data streams.

This improved proximity means that weaker but more power-efficient local communication protocols such as Bluetooth BLE can often be used by the sensors to send information[19].

Negatively, fog architectures are unsuited for highly isolated sensors outside the narrow ranges of these low-powered network protocols.

If you live in a major city and use the services of a global media streamer such as Netflix, chances are you're already benefiting from a type of fog computing, since Netflix is one of a limited number of tech giants with enough capitalization to create hardware caching installations within the infrastructure of the world's most popular ISPs.



The company's Open Connect program[20] is edge caching on steroids: according to demand, such portions of the media catalogue as are available to the host country are selectively copied over to a node *inside* the ISP's network — which means that the content you're watching is not being streamed from the internet at all but effectively from your own local network, similar to home-streaming applications such as Plex.

True to the fog paradigm, the 'local' Open Connect installation sends back collated data on user behavior and other statistics via AWS to a larger central server[21] in order to derive new insights for the company's extensive machine learning research program[22].

Edge IoT: Full-scale Local Analysis

In 'edge' IoT, the sensor or local equipment is powerful enough to process data directly, either through its on-board hardware or a hard-wired connection to a capable unit.

Predictably, power consumption and deployment and maintenance costs rise rapidly when an IoT unit becomes this sophisticated, making the edge IoT proposition a hard trade-off between cost and capability.

Edge computing is suitable for cases where autonomy is critical. This includes the development of self-driving vehicles, which depend on low-latency response times and cannot afford to wait for network interactions[23]; vehicle traffic monitoring[24]; SCADA and other industrial frameworks[25]; healthcare[26]; and urban and domestic monitoring[27], among other sectors.

In terms of deployment logistics and cost, it could be argued that edge computing is an effective return to the pre-IoT era of expensive and dedicated bespoke monitoring and data analysis systems, with a little periodic cloud exchange and machine learning added.

However, the long-term business appeal of AIoT may lie in a more creative use of existing resources, technologies and infrastructure, to enable AI-based technologies to operate effectively and *economically* at a local level.

When it comes to fog computing, Netflix is one of a limited number of tech giants with enough capitalization to create hardware caching installations within the infrastructure of the world's most popular ISPs.

AIoT: The Value of Client-specific Machine Learning Models

We saw in our recent article on Apple's Core ML that it's not only possible but preferable to train a machine learning model locally, so that it can adapt to the patterns and habits of a single user.

The alternative would be to constantly send user input data over the internet so that it can be processed by machine learning frameworks on a central server, and the derived algorithms sent back to the user's device.

This is a terrible approach for many reasons:

- It slows down inference analysis and performance in the user's device[28].
- It's not scalable or affordable, since a company such as Apple would then need to allocate expensive neural network processing time for the majority of its 1.4 billion mobile device users[29].
- In the case of high-volume providers, the volume and frequency of the transmitted data could compromise general internet capacity and latency times[30].
- It leaves the device 'stranded' outside of network coverage or in the case of a central server outage.
- It increases the attack surface of the data in transit[31,32], as well as the exposure of the user and the company.
- It's likely to infringe on the mobile data caps of many users, particularly in the US[33].
- While it may be useful for statistical customer analytics, the data is so specific to the user/client that, arguably, it has relatively little generalized value in terms of developing better machine learning workflows on a central server.

The Core ML infrastructure instead facilitates on-device machine learning templates across a range of sectors, including image, handwriting, video, text, motion, sound and tabular-based data.

Pre-Trained Machine Learning Models for Quick Deployment

It's much quicker to make a user-specific machine learning model from a template that's been partially trained in the intended domain (i.e. text inference, computer vision applications, audio analysis) than to train it up from zero.

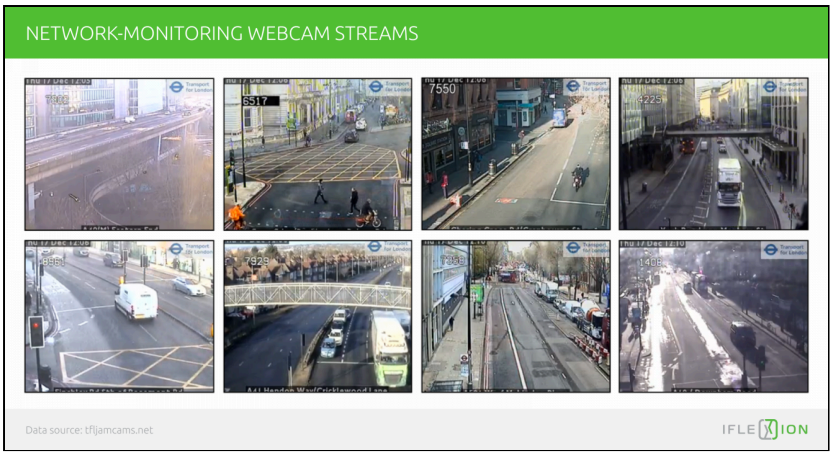
In the case of a local predictive text AI model, it's enough to provide a partially-trained template for the target language, such as English or Spanish. A neural network in an on-device System on a Chip (SoC) can then provide incremental text-based data from user interaction until the model has adapted completely to the host.

Text data of this kind might also include transcribed speech via AI-driven audio analysis — another 'tightly-defined' domain that's amenable to optimized and lightweight model templates.

Some Sub-Sectors of AI Are Easier to Generalize Than Others

In the case of video analytics, one of the most sought-after areas in IoT development, it's not quite so simple, because the visual world is much harder to categorize and segment than with narrow domains such as text.

Consider the huge variation in data from Transport for London's network of traffic-monitoring webcams:



The low-resolution feeds and sheer variety of shapes and fixed/periodical occlusions, combined with the peculiarities of weather and lighting, would defy any attempt to create a generic 'traffic in London' template model for deployment to an AI-enabled sensor network. Even if it were feasible, the 'base' model would need to be so comprehensive and sprawling as to preclude use in a lightweight AIoT environment.

Instead, an opportunistic AIoT approach would add local machine learning resources, either per-device or via a mesh network of nearby processing hubs, to help each installation become a 'domain expert' in its own particular fixed view of the world, starting from a minimally pre-trained model.

In such a case, initial training would take longer and require more human intervention (such as manually labelling certain frame-types as 'accident', 'traffic jam', etc. so that the device learns to send high-value alerts as a priority over the network noise); but over time, essential 'event patterns' would emerge to make early generalization easier across the entire network.

Dedicated AIoT Hardware Solutions

There is currently a proliferation of research initiatives to develop low-power ASICs, SoCs, and other integrated and embedded solutions for AIoT.

The most prominent market leaders at the moment are Google's Edge Tensor Processing Unit (TPU)[34], used in a famous match against Go world champion Lee Sedol[35], and NVIDIA's Jetson Nano[36], a lightweight AIoT quad core unit available from US$99.

Other commercial contenders and projects in development include:

- **Xcore.ai**

A highly configurable AI micro-processor[37] from British group Xmos, priced as low as US$1.

- **Cortex-M55**

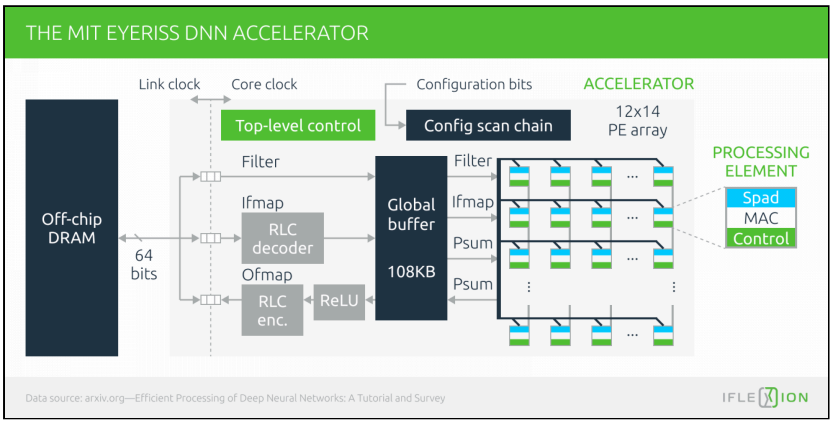
A very low-power ARM microarchitecture[38], now owned by NVIDIA.

- **Stanford's Energy Efficient Inference Engine (EIE)**

A new architecture to accelerate deep neural networks by compressing them into on-chip SRAM, achieving 120x energy savings over comparable networks[39].

- **The MIT Eyeriss Chip**

A deep convolutional neural network (CNN) accelerator[40] from MIT that takes advantage of the primitive generalization architectures of popular open source machine learning frameworks such as PyTorch, Caffe, and TensorFlow.



- **Intel Movidius Myriad X Vision Processing Unit**

Featuring a native 4K image processor pipeline and accommodation for eight high-definition sensors, the Myriad X[41] is aimed at computer vision development with an emphasis on minimal power requirements. The unit was adopted in 2017 by Google for its Clips camera[42].

- **Greenwaves Gap9 Wearable Processor**

Designed for battery-driven sensors and wearable devices, Gap9 trades performance against endurance and offers up to 32-bit floating point computation[43].

- **Mythic Intelligent Processing Unit accelerator**

Designed for vision-based and data center deployments, the Mythic processor uses an innovative approach to weight compression in order to fit machine learning operations into very lightweight deployments by utilizing flash memory[44].

- **Rockchip RK3588 SoC**

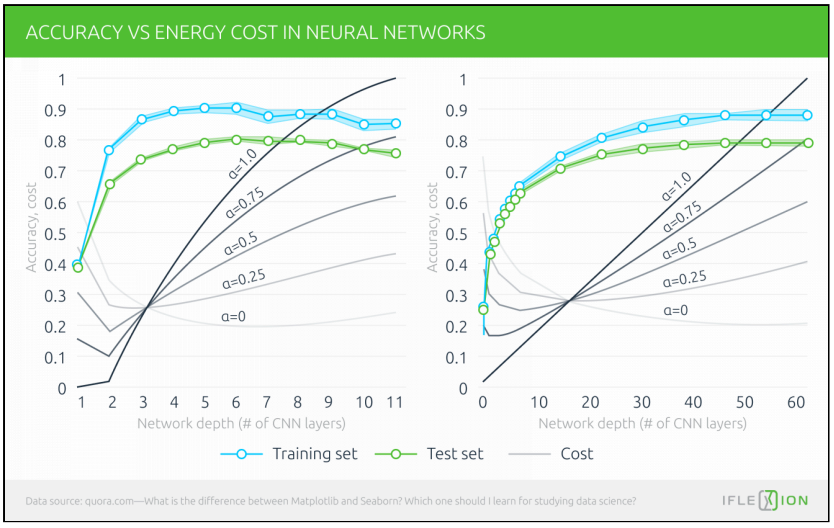
Chinese SoC manufacturer Rockchip will follow up[45] on the success of their RK3399Pro AIoT unit[46] with this multi-platform offering, which features an NPU accelerator, support for 8Kp30 video streams, eight processor cores, and 4x Cortex-A55 chips.

- **Syntiant NDP101 Neural Decision Processor**

Featuring a processor-in-memory design, NDP101 consumes less than 8 watts while performing inference weight operations inside Amazon's Alexa smart assistant range of products[47].

The Challenge of AIoT

The above are only a selection from an ever-widening range[48] of hardware and software architectures that are currently seeking to solve the performance vs. energy conundrum in AIoT.



The challenge of AIoT is also a spur to invention, since the sector's future depends on fully exploiting limited local resources to run low-power and highly targeted neural networks for inference and other local machine learning workloads.

While the evolution of AIoT hardware is faced with quite rigid logistical boundaries (such as battery life, limited local resources, and the basic laws of thermodynamics), research is only now beginning to uncover new software approaches to minimize the energy drain of neural networks while keeping them performant in low-resource environments[49].

This impetus is set to not only eventually benefit mobile AI, but to feed back into the wider machine learning culture over the next five or ten years.

The challenge of AIoT is a spur to invention. This impetus is set to not only benefit mobile AI, but to feed back into the wider machine learning culture over the next 5-10 years.