Mobile AI

Machine Learning/AI processing on mobile and IoT Edge Devices

Emilio Jose Coronado Lopez
http://seoulai.com/

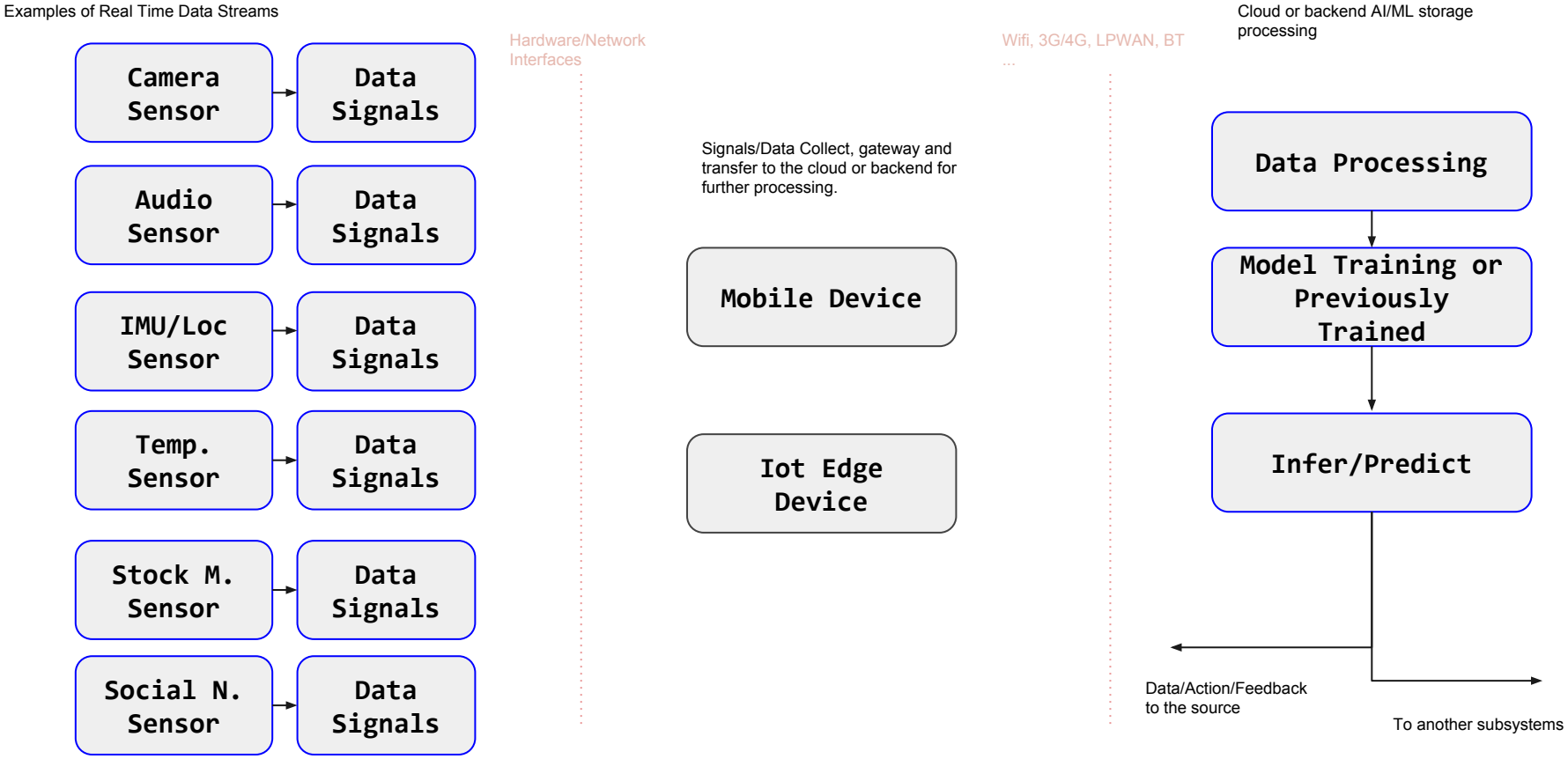
Paradigm

We’re living in a data world. A world of connected sensors, signals and data streams.

Sensory data continuously flows and gathers into a form of mobile or IoT Edge device, then processed, transferred through network to cloud backends, storage, analytics services...

Analytics, outputs coming from it, are some kind of offline, late, delayed processing, mostly runs on stored or historical data. This has been pretty much the main scenario of many business and entities entering into digital industry 4.0.

However, it will scale and growth through instant feedback, real time ML/AI services, IT and cloud providers are quite use to those incremental paths on controlled environments, but adding wireless, mobile networks, tough latencies and response times hard constraints beyond the confined system is another story.


Examples of Real Time Data Streams
Camera Sensor
Data Signals
Audio Sensor
Data Signals
IMU/Loc Sensor
Data Signals
Temp. Sensor
Data Signals
Stock M. Sensor
Data Signals
Social N. Sensor
Data Signals
Hardware/Network Interfaces
Signals/Data Collect, gateway and transfer to the cloud or backend for further processing.
Mobile Device
Iot Edge Device
Constraints: Latencies
Wifi, 3G/4G, LPWAN, BT ...
Cloud or backend AI/ML storage processing
Data Processing
Model Training or Previously Trained
Infer/Predict
Data/Action/Feedback to the source
To another subsystems
Constraints: Latencies, Processing, Bandwidth, data storage ....

It’s happening now

A typical big data, analytics pipeline integrated with Open Source machine learning frameworks like Tensorflow, Caffe, Keras, etc. enabling top notch applications and user experiences and putting all this collected data into actionable value in the backend.

Also commonly distributed computing SW and tools like Apache Kafka, Spark or Storm, open the real time streaming instant analytics capability tag into the box, so in “theory”, responses and actions can flow quick enough to any side of the network.

But is it “real”, real time ?

How real time is it?

Real time imposes hard timing constraints, depending on the use cases, it means you need sampling fast enough feed the analytics engine fast enough, but also means the system has to generate responses quick enough to keep up with a defined UX experience, or generate events on time for critical hardware requirements ...

Those actions and events needs to be delivered when and where are needed, maybe in the backend, maybe in the device that generates the data, or maybe somewhere else in the edge of the system who are listening.

Even there is enough processing power, network latencies, bandwidth and reliabilities are worst enemies to guarantee real time.

Examples

Online Predictions, Shopping assistants, Recommendation Systems: Not really real time systems, precalculates and generates results offline.

NLP, AI assistants, Low-res games: Audio/Video streaming not really hard real time small delays can be solved with “reasonable” buffering, and pre-processing techniques.

Smart Cameras, High-res games, VR: Lower tiers of FPS, and resolution is fine, however high speed, going high resolution and 4K/8K video will start to push limits.

Connected Cars, UAVs, Robots, Medical Devices: Elephant in the room, as those generate plenty of instant data, and require instant feedback, plenty of scenarios, have a lot of hard real time, regulated requirements.

Smart Farming: Some places even does not have decent or reliable connectivity to stay 24/7 online.

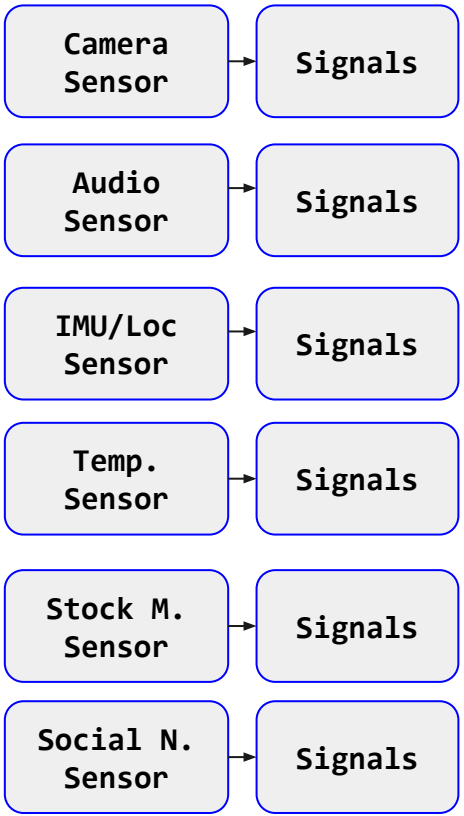
Enter AI/ML on ARM

On desktop and servers, NVIDIA is king https://www.nvidia.com/en-us/deep-learning-ai/

*99.99% of referenced mobile or edge IOT SOC based devices uses some kind of ARM CPU many of them paired with a GPU with OpenCL/OpenGL support.

AMD and Intel are trying to quick catch up, but who knows.

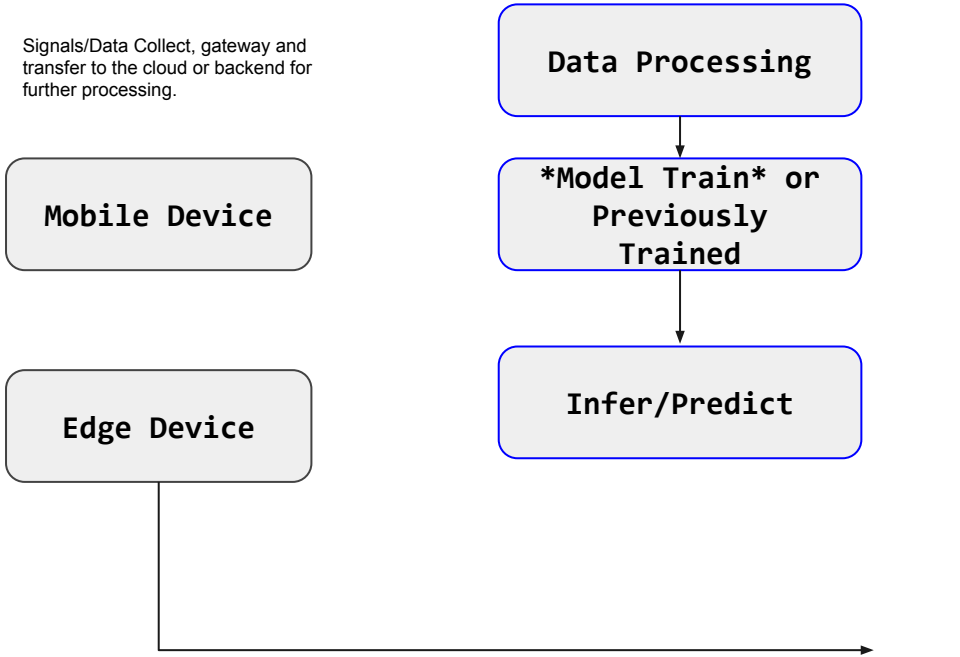
Examples of Real Time Data Streams



Constraints: Latencies

Signals/Data Collect, gateway and transfer to the cloud or backend for further processing.

Mobile GPU or attached Neural Compute Unit (Movidius)



Constraints: CPU/GPU Processing, Memory bandwidths, Batteries

ML/AI on the edge

In theory, moving some of the processing into the edge of the network will help with latencies, and response times...

Today, is a trade off, which data is valid to be processed out of the boundaries? What can be isolated of cloud services ? , what is really possible in terms of processing power, what other challenges introduce like security is a remain to be seen.

Constraints are mostly hardware: CPU, GPU speed, RAM, heat, batteries, but also the fact that some data management will become more distributed, less trustable and prone to security flaws, in fact is a totally different complicated scenario.

But as we are engineers, let’s focus and play with the hardware and software itself for a while.

ARM

https://www.arm.com/markets/artificial-intelligence

ARM made available ARM Compute Library for CV/ML last year.

The open source, makers community is starting to play and port some of the typical frameworks:
https://ai.stackexchange.com/questions/2854/ssd-or-yolo-on-arm

https://www.theverge.com/2018/2/13/17007174/ai-chip-designs-arm-machine-learning-object-detection

ARM Based new SOC chipsets

http://i.mediatek.com/p60

4 Arm Cortex-A73 2.0 GHz, 4 Arm Cortex-A53 2.0 GHz NeuroPilot AI tech.
The MediaTek's NeuroPilot SDK in P60 is compatible with Google Android Neural Networks API (Android NNAPI), and also supports common AI frameworks, including TensorFlow, TF Lite, Caffe, and Caffe2. That makes it easy for developers to quickly bring-to-market innovative AI applications. As a partner of the Open Neural Network Exchange (ONNX), MediaTek is working on bringing ONNX support to the chipset in Q2 2018 to provide developers with even more flexibility for designing AI-powered applications.

Google

Google provides mobile versions of Tensorflow: Lite and Mobile

https://www.tensorflow.org/mobile/tflite/ , https://www.tensorflow.org/mobile/

https://developer.android.com/ndk/guides/neuralnetworks/index.html,

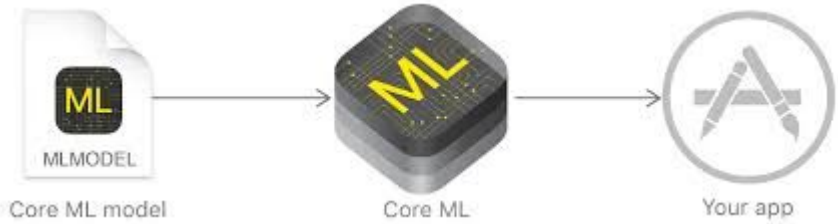
https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/android

Look into the Google TPU hardware, and Google Computer Engine too !!

https://cloud.google.com/tpu/

https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu

Apple



Apple has their coreML framework too:

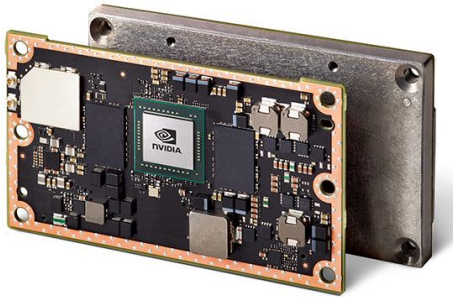
https://developer.apple.com/machine-learning/

https://www.udacity.com/course/core-ml--ud1038

Examples:

http://machinethink.net/blog/object-detection-with-yolo/

NVIDIA Jetson



https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/

NVIDIA Jetson is a no brainer choice for those who want to get serious about ML/AI development on “autonomous” devices.

I wrote it right “autonomous”, since for some reason NVIDIA never succeed in the mobile world.

The Jetson Development Pack (JetPack) includes tools for:

- Deep Learning: TensorRT, cuDNN, NVIDIA DIGITS™ Workflow
- Computer Vision: NVIDIA VisionWorks, OpenCV
- GPU Compute: NVIDIA CUDA, CUDA Libraries
- Multimedia: ISP Support, Camera imaging, Video CODEC

It also includes ROS compatibility, OpenGL, advanced developer tools ...

Intel

https://software.intel.com/en-us/ai-academy

That’s mostly using server and desktop CPUs and Intel SOC’s

But also bought/made interesting devices gives extra punch needed to crunch some of the usual ML/AI frameworks: Intel Movidius Neural Compute Stick

https://ncsforum.movidius.com/discussion/218/tiny-yolo-on-ncs

Finale

IOT, and new form of low latency, low power, constantly on sensors means more data points, signals, and streams to process in the gateways/edges

5G will connect more devices, demanding higher bandwidths, by instance 4K/8K, AR, VR streams.

Connected cards, and AUV will add millions of sensors feeds with real time requirements to backend and cloud services.

ML/AI processing will be splitted between on devices itself, and backends, cloud services.

finale

I recommend to start exploring with some hardware running Linux, widely supported by the community: Beagle Boards, Raspberry Pi's, ODroid's.

Use software architectures and solutions that can scale with mobile hardware chipsets developments.

If someone is interested, it seems Intel Movidius Neural Computer Stick is also available in Korea.

https://kr.mouser.com/new/Intel/intel-movidius-stick/

thank you.

Description:

Machine Learning/AI frameworks and pipelines are used to be executed or deployed in a kind of PC+GPU hardware solution. Recently Google, Amazon, Microsoft, Facebook, etc. are opening their internal AI infrastructure to the public, offering ML/AI cloud computing as a service, mostly running NVIDIA’s GPU’s or dedicated NCU (Neural Computing Units) in their servers.

This solution works pretty well on applications and services that does offline, analytics, defers computing, or does not have hard real time requirements, however, most of the mobile first applications, upcoming IOT, UAV applications, are pushing for more connected sensors, more data flowing into the cloud, and UX with none or close to zero latencies.

With more powerful mobile and embedded chipsets with specific GPU’s, AI/ML in mobile or IOT edge devices become possible, doing most or “enough” processing into the device itself, offloading some cloud computing processing and bandwidth for specifics.

This is a quick introduction, current SOC’s and platform status are probably not enough for the most advanced applications or uses cases, but those are first steps, there is already good open source and community support and is good enough to start playing with it.