libAPU: A path to upstream AI/ML accelerators

Alexandre Bailon

Artificial Intelligence and Machine Learning

Definitions

Artificial Intelligence

Artificial intelligence (AI) is intelligence demonstrated by machines, as opposed to the natural intelligence displayed by animals including humans.

Machine Learning

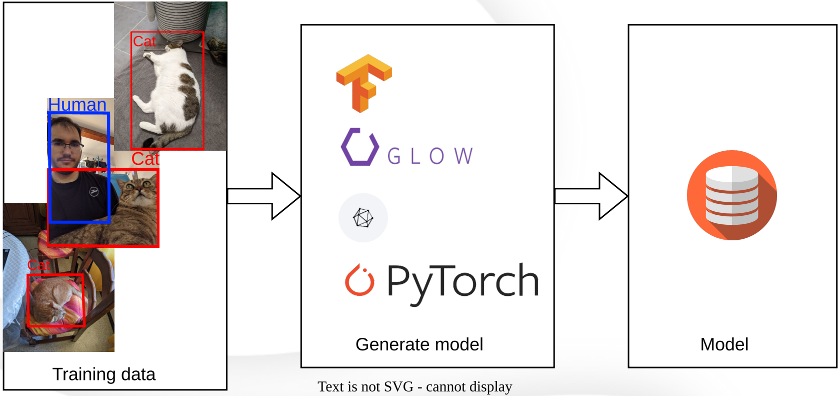
Machine learning (ML) is a field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks. It is seen as a part of artificial intelligence.

Definitions

Embedded Machine Learning

Embedded Machine Learning is a sub-field of machine learning, where the machine learning model is run on embedded systems with limited computing resources such as wearable computers, edge devices and microcontrollers.

Training a model



Training data

Generate model

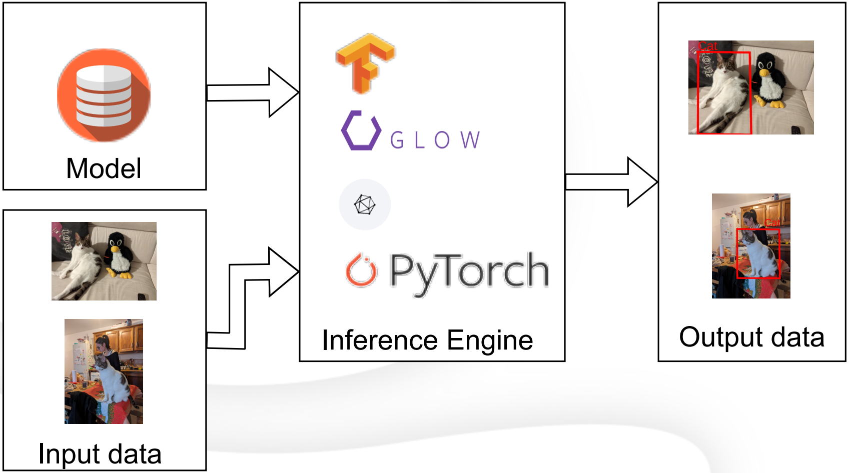
Model

Text is not SVG - cannot display

Training a model

- Prepare a dataset (pictures, text, audio, . . .)
- Annotate it (e.g. the coordinates of a cat in a picture)
- Train a model using your favorite AI/ML framework

Running the model



Model

Input data

Inference Engine

Output data

Running the model

- Start the inference engine
- Capture data and analyze them using the model and inference engine
- Post process the inference engine data (e.g. show where the cat is in the picture)

Running the model

- Start the inference engine
  - Load model in memory
  - Initialize the accelerator, load and start the firmware
  - Generate a graph of operations to run on the accelerator
- Capture data and analyze them using the model and inference engine
- Post process the inference engine data (e.g. show where the cat is in the picture)

Running the model

- Start the inference engine
  - Load model in memory
  - Initialize the accelerator, load and start the firmware
  - Generate a graph of operations to run on the accelerator
- Capture data and analyze them using the model and inference engine
  - share data with the accelerator
  - the accelerator unfold the graph and execute mathematical operations
  - the accelerator share the result with the CPU
- Post process the inference engine data (e.g. show where the cat is in the picture)

Embedded Machine Learning

A lot of software and frameworks

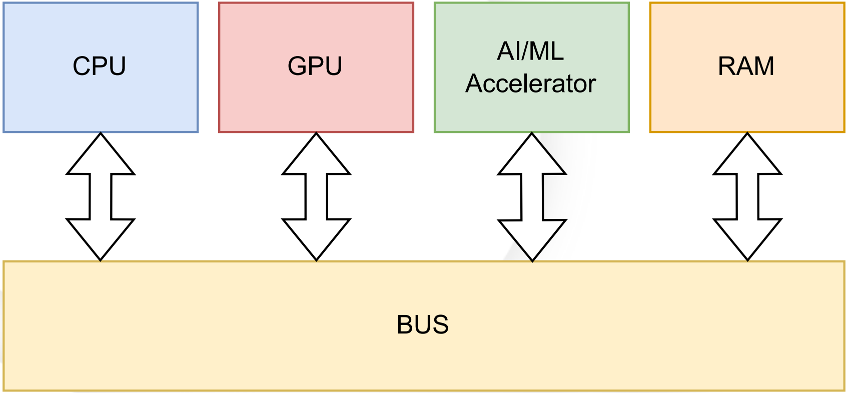
- Tensorflow
- Glow
- Caffe
- PyTorch
- ONNX
- TVM
- ...

Embedded Machine Learning

Wide variety of hardware accelerators

- GPU
- VPU, DSP or CPU (using additional instruction set)
- Neural Network accelerator

Embedded Machine Learning



Common hardware architecture

- Hardware accelerator(s) integrated in SoC
- Requires firmware to work
- Shared memory with CPU
  - CPU and accelerators can exchange data directly using shared memory

Common software architecture

- An inference engine
- A model
- Firmware has to be loaded on the accelerator
- A kernel driver to manage and communicate with the accelerator

Common software architecture

- An inference engine
  - with a vendor HAL:
    - to know the list of operators supported
    - to load models
    - to execute a model or individual operators
- A model
- Firmware has to be loaded on the accelerator
- A kernel driver to manage and communicate with the accelerator

Common software architecture

- An inference engine
- A model
  - Generated for an inference engine
  - Generated to support a specific operand type (float, int8, int16, . . .)
  - Model type (the algorithms and mathematics) used by the model
- Firmware has to be loaded on the accelerator
- A kernel driver to manage and communicate with the accelerator

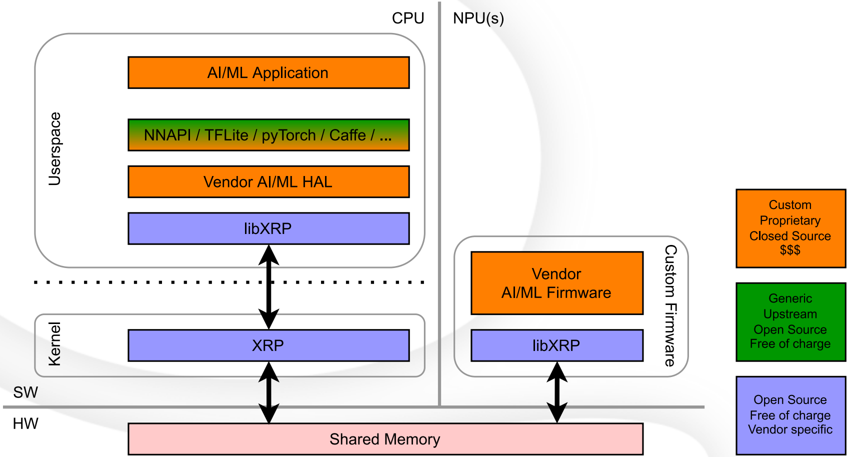
Common software architecture

- An inference engine
- A model
- Firmware has to be loaded on the accelerator
  - Built using closed source toolchains
  - Implements mathematics operations optimized for the accelerator
  - Handle CPU commands
- A kernel driver to manage and communicate with the accelerator

Common software architecture

- An inference engine
- A model
- Firmware has to be loaded on the accelerator
- A kernel driver to manage and communicate with the accelerator
    - Power up the accelerator
    - Load the firmware
    - Provide communication layer between CPU and accelerator
    - Manage memory

Example: Cadence XRP



CPU
NPU(s)
Userspace
AI/ML Application
NNAPI / TFLite / pyTorch / Caffe / ...
Vendor AI/ML HAL
libXRP
Kernel
XRP
Vendor AI/ML Firmware
libXRP
Custom Firmware
SW
HW
Shared Memory
Custom Proprietary Closed Source $$$
Generic Upstream Open Source Free of charge
Open Source Free of charge Vendor specific

Common issues

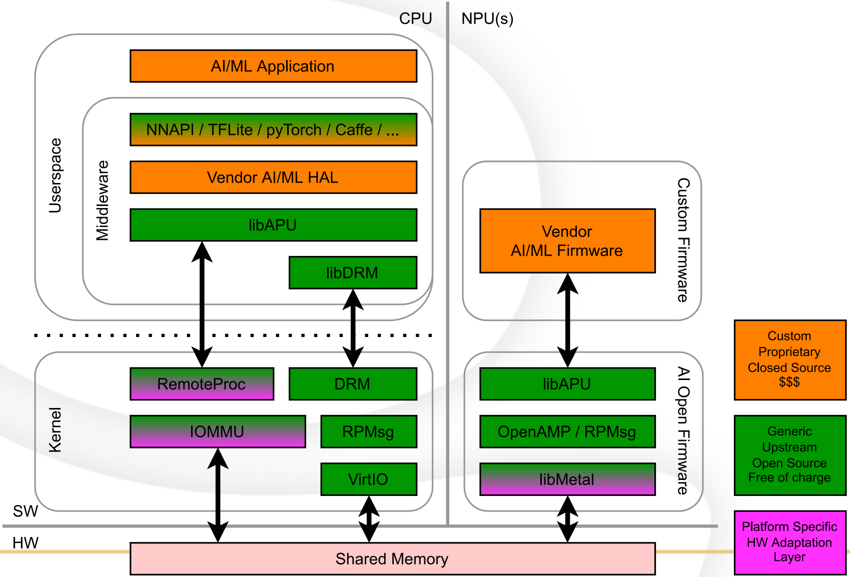
- Firmware
  - Usually made for one or few inference engine
  - Usually closed source, built using closed source toolchains
  - Use in house RPC to talk with CPU
- Kernel and userspace
  - Sometime, almost only userspace
  - No code reuse (re-implement a lot of existing features)
  - Use in house RPC solution
  - Vendor specific

libAPU

A first step to leverage Machine Learning support

- Implement a common RPC stack
  - Generic, could be run on many accelerators
  - Only requires a shared memory between CPU and accelerators
  - Open source, could be used by firmware and inference engine
  - Relies on existing open source solutions
    - OpenAMP / RPMsg for firmware
    - DRM, RPMsg, remoteproc, dma-buf

libAPU Architecture



Overview

- Userspace library to control and manage messaging between AI/ML application(s) and AI/ML HW accelerator(s)
  - HW Accelerators power management (on/off/sleep)
  - Firmware download
  - Messaging
  - Memory Allocation
- NB: does not implement any mathematical or AI/ML functions

Features

- RPMsg-based generic messaging
  - Use Shared Memory
  - Handle both synchronous and asynchronous requests
  - Multi-Channel (to be added)
- RemoteProc-based generic power & firmware management
- Multi / Heterogeneous Core Support

Features

- DRM-based Task scheduler
    - Send RPC Messages to unused / requested cores
- DRM-based Memory management
    - Contiguous memory, user buffers
- Configured using device tree

Status

Status

- In production
  - libAPU integrated with Cadence XANN (NNAPI / tflite) on MTK i350/i500 architecture
- libAPU RFC under review
  - RFC submitted in September 7th, 2021
  - MTK i350/i500 architecture used as reference platform
- Performanceson par with proprietary solutions

NVDLA (NVIDIA Deep Learning Accelerator)

- NVIDIA upstream support for it accelerator
  - https://lwn.net/Articles/891865/
- Also use DRM
- But only target NVLDA
  - libAPU relies on Remoteproc to manage hardware
  - But also highlight that we may have to add HAL to use DMA / SRAM

What’s Next ?

- Get libAPU adopted upstream
- Add libAPU support for other platforms / architectures
  - TI AM5729 and its 4 Embedded Vision Engines (EVE) might be a good target
- Add multi-channel support
- Performance improvements
  - Zero copy, . . .
- Design a libAPU Open Firmware Architecture for AI/ML co-processors
  - Similar to Intel’s Sound Open Firmware (SOF)

Contribute

Kernel

- https://lwn.net/Articles/869547/

libAPU

- https://github.com/BayLibre/drm/tree/apu-support
- https://github.com/BayLibre/open-amp/tree/v2021.10-mtk
- https://github.com/BayLibre/libmetal/tree/v2021.10-mtk

Thank you