State of the tinyAutoML Market **2022**

June 10, 2022

Introduction

Tiny machine learning is broadly defined as a fast-growing field of machine learning technology and applications including hardware and software capable of performing on-device sensor data analytics at extremely low power. The tinyML Foundation is powering the explosive growth of this ecosystem.

tinyAutoML is the process of automating tiny machine learning applications. Automation in this process allows non-experts to access the capabilities of ultra-low-power at the edge applications as well as increasing the productivity of experts.

In the first half of 2022, the tinyML Foundation started a working group focused on the Auto ML area within tinyML. That working group resulted in this document as well as the tinyML Auto ML Forum on June 15, 2022. The remainder of this document consists of the brief summaries each company provided of their tinyAutoML tools, following a common template, intended to share with users:

- What is the tool and what does it do?
- What are the tinyML use cases and what types of data does the tool support?
- What are the parts of the tinyML product development process that the tool helps with?
- How does the tool impact the customer's productivity?

We hope that this document demonstrates our excitement for tinyML and specifically tinyAutoML and the impact it will have on the productive deployment of intelligent internet of things and ongoing training of machine learning models on edge devices. We encourage you to contact each of the companies represented here to learn more about their exciting tools and this dynamic industry.

tinyML Auto ML Technical Program Committee
Elias Fallon, Chair, Qeexo
Danilo Pau, STMicroelectronics
Davis Sawyer, Deeplite
Martin Croome, GreenWaves
Kate Vasilenko, Neuton
Sam Al-Attiyah, Imagimob
Tomas Uppgard, Stream Analyze
Evgeni Gousev, Qualcomm Research, USA
Rosina Haberl, tinyML Foundation
Ira Feldman, tinyML Foundation

Note: The order of companies in the remainder of the document was randomly determined (using np.random.shuffle, contact Elias for the code) and has no specific significance.

Edge Impulse / EON Tuner

Jan Jongboom, jan@edgeimpulse.com

Introduction

The EON Tuner helps you find and select the best embedded machine learning model for your application within the constraints of your target device. The EON Tuner analyzes your input data, potential signal processing blocks, and neural network architectures - and gives you an overview of possible model architectures that will fit your chosen device's latency and memory requirements. The EON Tuner is part of Edge Impulse (https://studio.edgeimpulse.com to sign up) and is available for all users.

Docs: https://docs.edgeimpulse.com/docs/edge-impulse-studio/eon-tuner
Intro and video:
https://www.edgeimpulse.com/blog/introducing-the-eon-tuner-edge-impulses-new-automl-tool-for-embedded-machine-learning

Use Cases and Data

The EON Tuner can be used for any type of sensor data that is supported by Edge Impulse, and comes with presets for accelerometer, audio and vision models; for which you can solve classification, object detection and regression tasks. In addition, the EON Tuner has a fully customizable search space and the ability to plug in custom DSP and ML blocks - making it applicable to any sensor data and model architecture.

An important feature of the EON Tuner is that it understands the device constraints and outputs models that will fit the latency and memory requirements of the application. Models that don't fit these constraints will be evicted from the search space. The models it outputs can be ran anywhere from the smallest MCUs to specialized DSPs, NPUs and even MPU + GPU (delivered as C++ source code).

Process

The EON Tuner integrates deeply with the other parts of the Edge Impulse Studio covering the complete Edge ML workflow. This includes easy data acquisition from a wide variety of devices, direct integration with customer's data lake [1] and data exploration and data labeling with the Data Explorer (which uses neural network embeddings to give insight in your data) [2].

Once you have an interesting dataset you can either model your application using a wide variety of high performance DSP and ML blocks; use the EON Tuner to help you find (or finetune) a

new model architecture; or plug in custom preprocessing, DSP and ML blocks to give you full freedom in designing your models.

After building your model the Studio lets you test your model against up to 24 hours of real-world data to get accurate performance characteristics, showing the strong and weak points of your model; and choose post-processing parameters to optimize for a low False Reject Rate (FRR) or False Activation Rate (FAR) - giving confidence in your models.

Models are deployed as royalty-free C++ source code without any external dependencies, and come with hardware acceleration for a wide variety of platforms incl. general purpose MCUs, MPUs, specialised NPUs and even some GPUs.

Once deployed you can use datalake integration or the ingestion APIs to automatically send anomalies or samples that you're unsure of back to Edge Impulse for manual review (assisted by the Data Explorer).

[1] https://docs.edgeimpulse.com/docs/edge-impulse-studio/data-sources
[2] https://docs.edgeimpulse.com/docs/edge-impulse-studio/data-explorer

Productivity

The best quote we got from an end customer was that they realised they no longer needed any machine learning, after finding the best set of signal processing parameters using the EON Tuner. A simple `switch` statement was enough to determine all the states they needed.

The development community has seen tremendous improvements in model accuracy without performance penalties, as the EON Tuner lets you evaluate hundreds of DSP+ML combinations in one go, something that's very hard to do by hand. E.g. here's an example of going from 82% => 92% accuracy in one run:
https://create.arduino.cc/projecthub/dhruvsheth_/eon-tuner-automl-in-embedded-ml-with-edgeimpulse-sony-5546d1

Summary

Don't think of AutoML as some silver bullet, but rather as an engineering tool. It definitely can show you interesting model combinations with the press of a button, but you'll get the best results if you have a strong understanding of the problem you're trying to solve, and thus can evaluate the quality of your dataset and even do custom feature engineering work. Then use a tool like the EON Tuner to evaluate all potential combinations quickly.

Martin Croome <martin.croome@greenwaves-technologies.com>

Introduction

GreenWaves Technologies is a fabless semiconductor startup that designs and brings to market advanced ultra-low-power AI and DSP processors for energy-constrained applications. As part of our GAP Software development Kit we provide a number of different tools, which we call GAPFlow, that automate the process of transforming neural network graphs from training tools such as TensorFlow and Pytorch into optimized C code that can be run on GAP processors. The GAPFlow tools allow a neural network to be converted, compiled and run directly through an easy to use Python API integrating perfectly with normal machine learning development processes.

Use Cases and Data

GAPFlow supports a wide range of different neural network architectures processing data such as images, sounds or radar signals. It includes full support for Convolutional, Transformer or Recurrent networks including SSD based object detection networks. GAPFlow's highly flexible quantization tools support mixing 16 bit float and 16 to 2 bit fixed point activations and parameters in a single network.

Process

GAPFlow can be used to port, evaluate and validate neural networks on GreenWaves GAP processors. Since GAPFlow is easy to integrate into any Python based project and can execute converted networks on GAP hardware or simulation platforms it can be used as part of a continuous integration process supporting regression testing in a production environment. It is also a vital tool during development and optimisation of embedded Neural networks.

Productivity

GAPFlow reduces the time to go from a trained floating point network in TensorFlow and PyTorch to a highly optimized C implementation ready for execution on GAP processors from weeks to hours. The ability of GAPFlow to be used inside a continuous integration process cuts down on expensive deployment mistakes. GAPFlow's extensive operator support reduces time wasted modifying networks for embedded systems.

Summary

GAPFlow is an end-to-end set of development tools for porting neural networks to GAP processors. It is available for download, free of charge, from:

https://github.com/GreenWaves-Technologies/gap_sdk

Intelligent Agent
Neuton

Neuton TinyML

Make your edge devices intelligent

G2 ★ 4,9 / 5 | Capterra ★ 4,9 / 5 | Product Hunt #5 Product of the Day

Meet **Neuton TinyML**, a no-code tiny machine learning platform that empowers users to build extremely compact models and embed them into unbelievably tiny devices, even with 8-bit capacity.

Having a **unique neural network framework** under the hood, Neuton automatically creates models of minimal size and without loss of accuracy:

- silicon agnostic
- with fewer coefficients and neurons
- up to 1,000 times smaller compared to other frameworks

Check out our benchmarks here.

Deploying innovative tinyML solutions on edge devices shouldn't cost a fortune, so we provided our platform with an **absolutely free unlimited plan**.

REAL-WORLD USE CASES

Applicable for the following types of tasks:

- regression
- classification
- anomaly detection

Native embedding into:

- 8-bit
- 16-bit
- 32-bit MCUs

Supporting the following types of data:

- sensor
- audio
- tabular

Use supervised learning to perform the most common tiny machine learning tasks such as:

- create **smart interfaces** based on voice, gestures, or movements
- perform **predictive maintenance** of vehicles or equipment

- **monitor conditions** and **track assets**
- **recognize speech**, commands, and wake words
- **classify sound events**

Explore how to monitor water pollution, prevent water pump failure, detect people in the room, and even predict the future champion of Formula One in the practical use cases implemented by our community. You can find the full collection of projects **here**.

TinyML PIPELINE

Neuton's highly automated and transparent pipeline doesn't require a lot of actions from the user's side:

Upload your data in a CSV format

Neuton automatically performs digital signal preprocessing, selects the most relevant features, and creates an optimal model

Download the ready-to-use C library and embed it into a microcontroller

Run inferences on the device

Use Neuton's Explainability Tools:

- to evaluate quality (exploratory data analysis, model quality diagram)
- to interpret the results (model interpreter, feature importance matrix)
- to monitor performance (model-to-data relevance indicator)

WHY NEUTON TinyML?

Speaking of the tangible benefits of using Neuton, just imagine that now users of any tech level can:

create AI-driven IoT and edge devices **without**:

- any data science knowledge
- model compression techniques
- search for neural network parameters

produce incredibly small models **<1Kb** and run inferences faster

improve productivity by building models in just **3 clicks** and accelerate time to market

Discover how to apply the best tinyML practices to automatically create and deploy a super compact and accurate model with Neuton TinyML **absolutely free!**

Contact us at welcome@neuton.ai

STREAM ANALYZE

Tomas Uppgård, tomas.uppgard@streamanalyze.com

Magnus Gedda, magnus.gedda@streamanalyze.com

Introduction

Stream Analyze provides an end-to-end platform for development of edge AI solutions. Our platform combines the capabilities of real-time streaming analytics and on-device edge AI. By this we provide unique benefits in accelerating the processes of bringing edge AI solutions to market. The platform is easy to learn and use for data scientists, engineers, domain experts and other groups without deeper coding skills. Stream Analyze works mainly with large Industrials and Automotive companies, enabling a broad range of use cases.

Use Cases and Data

The platform is generic and enables a multitude of different use cases utilizing a wide spectrum of sensors. A broad set of model categories are supported to cater for different use cases. Different categories of models can be combined for more advanced use cases.

Types of Data/Sensors

- Wide spectrum of sensors (and data types) supported such as, accelerometer, gyro, microphone, temperature, humidity, gas
- Easy for users to add support for more sensors and data types
- Vision support is in alpha stage

Types of Machine Learning Algorithms supported

- Support for 1000+ mathematical and statistical functions for basic analytics
- ML, examples: Random Forest, DBSCAN, DenStream, k-NN, k-Means
- NN: TensorFlow Lite, Proprietary Inference Engine
- Platform is extensible with more models. Pretrained models can be imported and used in the workflow. Ongoing work to support models utilizing HW acceleration.

Target Platforms and/or Devices

Our solution works for different combinations of processor architectures and OSes including MCUs, typically ARM Cortex M based devices and “larger”. We usually port our solution to the devices/MCUs used by our customers.

Supported platforms:

- Generic: Windows, OSX, Linux, Android

- Device examples: NXP IMX 6 and 8, Microchip SAMA5D27, HMS Anybus, Raspberry Pi, MX-4 T30

Special Use-Cases or Verticals
We focus mainly on large Industrials and Automotive companies. Examples of use cases are intelligence in products enabling new services, analysis of product usage for product development, improved operations, predictive maintenance, enabling new service-based business models and more.

Process

Stream Analyze provides an end-to-end platform enabling efficient and scalable processes to bring edge AI solutions from idea to full scale deployments. Here is a summary of the key functions/steps in the process that we support:

- Enable real time data streams on edge devices and microcontrollers
- Powerful query and filtering capabilities to filter out exactly the data streams needed for the use case at hand
- Develop and apply computations and models to data streams
- Deploy models to edge devices instantly (bypassing the need for FOTA updates)
- Integrated tools for orchestration enabling management of portfolios of models across large fleets of industrial products and equipment
- Real-time visualization of data streams and model results
- Integration with other IT systems

Productivity

We enable a very fast process to bring edge AI solutions from idea to market. A complete iteration cycle including data filtering, modeling and deployment can be done in a few minutes compared to days, weeks or months, using traditional methods.

Summary

We provide an edge AI platform tailored for the needs of large industrial companies aiming to develop a multitude of solutions across large fleets of connected products and equipment.

Learn more and try our platform!
Intro video: https://www.dropbox.com/s/7pypoyhsuh6rwpv/Stream_Analyze_promo.mp4?dl=0
Product video: https://www.dropbox.com/s/kppunk8tt3kgp5h/SA_Engine_promo.mp4?dl=0
Free Community Edition: https://studio.streamanalyze.com

Google / QKeras & Vizier / YouTube Codec ML Team

Daniele Moro (danielemoro@google.com)

Introduction

QKeras is an open source quantization extension to Keras that provides drop-in replacements for quantizing Keras layers. It features a rich and flexible authoring API that enables users to custom-fit quantized models for even the most restrictive hardware platforms. Through state-of-the art quantization-aware training algorithms and comprehensive quantization functions, QKeras enables users to achieve maximum model performance even when compressing to low-bit precision.

AI Platform Vizier is a black-box optimization service that helps you tune hyperparameters in complex machine learning models. Due to the flexibility QKeras offers in quantizing a model, a huge number of hyperparameters must be adjusted to create Pareto-optimal models. Vizier enables users to search these hyperparameters in a massively-parallelizable way, with easy integration with Google Cloud for automatically starting workers to search quantization configurations and model architectures. Through the use of published black box bayesian search algorithms, Vizier efficiently finds models to optimize over multiple objectives.

Use Cases and Data

QKeras and Vizier can be used to enable efficiency gains for quantized models towards essentially any target use case, dataset, or platform. QKeras must be used with models created using TensorFlow's Keras library, but any existing TensorFlow training algorithms are compatible. QKeras can be used to create classification, regression, or any other class of machine learning model, both supervised and unsupervised. There are restrictions to the types of layers that QKeras supports, but one can create custom layers, and we are always adding new QKeras layers as well. Vizier enables search of the huge number of possible quantization configurations and model architectures, so it is also compatible with any QKeras model.

The most powerful way to deploy a trained QKeras model is by creating a custom ASIC or deploying to an FPGA through the use of High Level Synthesis. Although there are numerous ways to achieve this, the most popular method is described in our paper published to Nature in collaboration with CERN, which you can find at this link. Beyond this, users may be able to convert the QKeras model to C++ code or a TFLite model, but these methods of deployment are still under development & exploration.

We have successfully used QKeras for a number of applications, and you can see published use cases here: detecting particle collisions, classifying images with binary models, anomaly detection, home appliance classification.

Process

QKeras and Vizier can be used together to improve the tinyML Process from Model Development to Model Training. QKeras enables users to develop quantized models, and Vizier enables users to search for the optimal configuration and architecture of such quantized models. This can also be highly automated due to Vizier's integration with Google Cloud, and QKeras' advanced quantization-aware training algorithms that find optimal quantizer scales and activation calibrations.

Productivity

QKeras enables users to develop heterogeneously quantized models that were impossible or infeasible to implement using prior techniques. Vizier enables mass parallelization of quantization configuration searches, increasing productivity by orders of magnitude depending on the number of concurrent workers / machines that can be spun up at the same time.

Summary

QKeras and Vizier are a powerful combination, enabling users to develop quantized models that are far smaller and more accurate. QKeras allows for state-of-the art heterogeneous quantization-aware training, while Vizier augments the user experience with massively parallelizable black box bayesian optimization of hyperparameters. This enables users to develop models that are so small, fast, and accurate that they could be deployed on the edge in particle colliders.

Start using QKeras here: https://github.com/google/qkeras

Start using AI Platform Vizier here: https://cloud.google.com/ai-platform/optimizer/docs/overview

Optionally, you can also use the in-development open source version of Vizier here: https://github.com/google/vizier

For a deeper dive, please refer to this talk: https://www.youtube.com/watch?v=9A7jfpBsvO0

SensiML

https://sensiml.com
email: info@sensiml.com

Introduction

SensiML offers cutting-edge end-to-end AI workflow software for a wide array of IoT edge applications using time-series sensor data. With SensiML's Analytics Toolkit, developers can benefit from automated AI development software which includes data collection, labeling, feature extraction, ML classification and AutoML code generation. The result is drastically reduced development time and cost, allowing project teams ranging from single users to large teams to generate optimized edge AI sensor algorithms in a fraction of the time that would have otherwise been required with hand-coding.

SensiML sets itself apart with ML DataOps tools that support a host of critical dataset development tasks such as flexible sensor data acquisition, dataset importation, version control, video annotation, automated labeling, and detailed metadata annotation. Since the average ML project team expends 80% of its effort in data collection, labeling, augmentation, and cleansing, SensiML's Data Capture Lab was developed to streamline the *entire* model building effort, not just the ML modeling and code generation tasks of AutoML tools. The result is faster development with higher quality results, and improved accuracy.

Use Cases and Data

With over a decade of experience developing AI code for the extreme IoT edge, SensiML can help embedded developers add autonomous sensory intelligence to a wide array of IoT applications whether they be industrial sensors, wearable devices, consumer smart home, agricultural sensing, or smart building and infrastructure sensors.

Supported sensors include IMU (accel/gyro/mag), microphones, voltage, current, loadcells, strain gauges, barometric pressure sensors, gas and flow sensors, piezoelectric vibration, EKG/EMG/EEG electrodes, PPG, PIR and PIR arrays, radar, LVDTs, and other time-series sensors.

SensiML has a broad array of partners and supported platforms with out-of-the-box support for development kits from Arduino, Arm, Bosch, Broadcom, Infineon, Intel, Microchip Technology, Nordic Semiconductor, NXP, Qualcomm, QuickLogic, Renesas, Silicon Labs, Spark Fun Electronics, and ST Microelectronics.
SensiML's AutoML modeling engine covers a broad set of ML classifiers and can auto-optimize to fit hardware ranging from multi-core 64-bit application processors to 8-bit microcontrollers.

Supported algorithms include Neural Networks, TensorFlow Lite for Microcontrollers, SVM, k-NN, RBF, Decision Trees, and ensemble models.

Process

SensiML assists the user from start-to-finish throughout the ML model development workflow:
Device Setup: Library of pre-tested raw data collection firmware and board configurations
Data Collection: User configurable automated data capture with synchronized video option
Data Labeling: ML label automation with bulk file editing, auto labeling and session control
Data Management: Full client/cloud dataset sync and management for 1 to 100+ user teams
Model Setup: Library of templated common use cases with predefined model parameters
Pre-Processing: Simple to configure data filtering, triggering, and segmentation stages
Feature Engineering: Auto feature selection and optimization from library of 80+ features
Model Selection: Automated evaluation of range of classifiers without user expertise needed
Code Generation: Automated reduction of ML model to working binary, library, C source code
Code Optimization: Platform specific optimizations for CMSIS and vendor specific accelerators
Model Testing: Supports automated bit-exact model emulation testing in the cloud
On-Device Test: Streamlined target testing/logging with live data or full autonomous ML output
Continuous Learning: Augment datasets using on-device models and tuned personalization

Productivity

The SensiML tools have been proven to accelerate AI expert user productivity by at least 500%. Novice users can expect much higher productivity multiples afforded by the simplified graphical user interface and AutoML capabilities that eliminate the need to interact with complex programmatic open-source AI frameworks with their associated steep learning curves.

Summary

Sign up for a free SensiML Community Edition at https://sensiml.com/plans/community-edition and experience the benefits of SensiML's highly automated and simple to use TinyML software development toolkit. For more information, visit https://sensiml.com and check out our demos, tutorials, and detailed documentation.

Imagimob - Imagimob AI

Sam Al-Attiyah - sam@imagimob.com
Anders Hardebring - anders@imagimob.com

Introduction

Imagimob AI is a development platform designed to streamline the process of creating production-ready ANN models for time-series data to be run on devices with constrained resources. It's built on Imagimob's experience of enabling AI and ML functionality for different customers and products. The same tool we use ourselves for building these models we have opened up to the public, everything we have found to be useful we believe the customer will as well. Imagimob AI is low code but gives the user all the tools they need to go from a sensor connection to a model that is simple to integrate and completely ready with the associated pre-processing.

Use Cases and Data

Data/Sensors - Imagimob AI works with any type of **time-series data** and supports **CSV** format
Algorithms - Imagimob AI focuses on **supervised** and **semi-supervised, classification**, **regression** and **anomaly detection** problems using Artificial Neural Networks (**ANN**).
Model Formats - Imagimob AI produces models in a standard **H5** file format but it also has a transpiler allowing you to convert models to standard **C code**. This means it's very easy to integrate the models in **python** or **C** as the user prefers. We support quantization of LSTM layers which is supported by few other platforms, and we support deep learning predictive maintenance using autoencoders.
Target Platforms - Models built using our platform have been deployed on low power MCUs such as ARM Cortex M23 and some more powerful ones as the Cortex-M4 as well as ESP32. We have deployed on Synaptics DBM10L and are working with Syntiant to deploy on their NDP120. We have also deployed on Raspberry Pis and small PCs
Special Verticals - Our key verticals are gesture control using radar sensors, human motion including fall detection using IMU's, sound event detection/audio applications and predictive maintenance. We have done extensive work using radar technology such as building the world first gesture controlled earbuds and fall detection system. We also work with different industrial customers on predictive maintenance products that utilize different time-series sensors.

Process

1. **Collection and Acquisition** - support for different sensors/data types exists for streamlined data collection and all time-series data is supported in CSV format
2. **Analysis and annotation** - data visualization and easy annotation is available

3. **Preparation and Pre-processing** - provides statistical analysis of data distribution between sets and classes, also shows dimensionality and frequency of data. Many kinds of pre-processing are supported such as mathematical functions and fourier transforms. All are python and C compatible
4. **Training and Evaluation** - generate different architectures that are intended for edge deployment, the different architectures include dense, convolutional and LSTM/GRU.
5. **Conversion to Edge** - Can convert all layers used to Imagimob AI to C either quantised or un-quantised and also can convert models built outside of the studio either in H5 or ONNX. Single source file and 3 API function calls
6. **Deployment** - Integrated deployment process for selected chips and MCUs such as Arm Cortex M-series based MCU's, Synaptics DBM10L, Syntiant NDP120, various Texas Instruments chips and radars and Renesas RA2L1, amongst others
7. **Maintaining, iterating and continuous learning** - in development we are working on automating the continuous learning process and have a custom deployed solution in the field where data from false positives can be sent back for further model improvement to ensure continuous improvement

Productivity

Imagimob AI improves productivity in the following ways:

- **Ease of data collection** - all time-series data and sensors are supported in CSV format but for selected sensors we have further simplified to process of collecting data allowing you to plug and play certain sensors and save **weeks** of development time to prepare them or get them into the right format
- **Data annotation** - our combination of automatic labeling tools, ability to add metadata and easy to use annotation helps to half the time that you spend on the process which can save **days** or **weeks** depending on the amount of data
- **Data management**: our data quality verification saves **days** of development time. It ensures that your data is consistent in terms of data configuration such dimensionality and frequency which otherwise be a common problem when dealing with lots of data
- **tinyML AutoML** - saves **weeks** by ensuring that right when you start model building your models are edge compatible and every model is ready for deployment
- **Deployment** - easy to use conversion means that getting a model on the edge is much quicker. Only 1 file means that not only do you save **weeks** of work in the model conversion but also **days** fiddling with different libraries and files.

Summary

Imagimob AI is the platform to use if you want to build production ready models. Find out more about our offerings here or contact Alina, our tinyML solutions manager for a demo here.

Qualcomm

AI Model Efficiency Toolkit (AIMET)

Chirag Patel cpatel@qti.qualcomm.com. Principal Engineer/Mgr.

Introduction

Manual optimization of a neural network for improved efficiency is costly, time-consuming, and not scalable. The AI Model Efficiency Toolkit (AIMET) solves this challenge by offering state-of-the-art quantization and compression techniques that allow users to get memory, compute, and energy-efficient models for fixed-point inference while maintaining accuracy comparable to floating point models.

AIMET: https://github.com/quic/aimet , Demo: https://www.youtube.com/watch?v=QZjsCVx79_k; AIMET Model Zoo: https://github.com/quic/aimet-model-zoo;

Use Cases and Data

AIMET can be used to obtain INT8, INT16, or even mix-precision models that are significantly smaller in size compared to FP32 models. AIMET is versatile and proven to optimize models for popular as well as challenging use cases including classification, object detection, super-resolution, speech to text, and more.

Process

Users can incorporate AIMET's techniques into their PyTorch and TensorFlow model-building pipelines for automated post-training optimization, as well as for model fine-tuning (Quantization Aware Training (QAT)), if required.

Productivity

Automating model quantization and compression algorithms helps eliminate the need for hand-optimizing neural networks that can be time consuming, error prone, and difficult to repeat.

Summary

Use and contribute to AIMET at: https://github.com/quic/aimet

AIMET and AIMET Model Zoo are products of Qualcomm Innovation Center, Inc.

Qeexo AutoML

Leslie J. Schradin, III, leslie@qeexo.com Principal ML Engineer
Michael Gamble, michael.gamble@qeexo.com
Director of Product Management

Introduction

Qeexo AutoML is a fully automated, end-to-end, machine learning platform providing users with the ability to rapidly collect & edit sensor data, train & evaluate models and metrics, and deploy & live test up to 17 different algorithms without writing a single line of code. Built for speed and efficiency, Qeexo AutoML's NO CODE platform reduces time and dependencies on expensive expert resources, drastically simplifying deployment and increasing scalability. Simple and intuitive, Qeexo AutoML has been designed to support the needs of machine learning practitioners, embedded engineers, and domain experts alike.

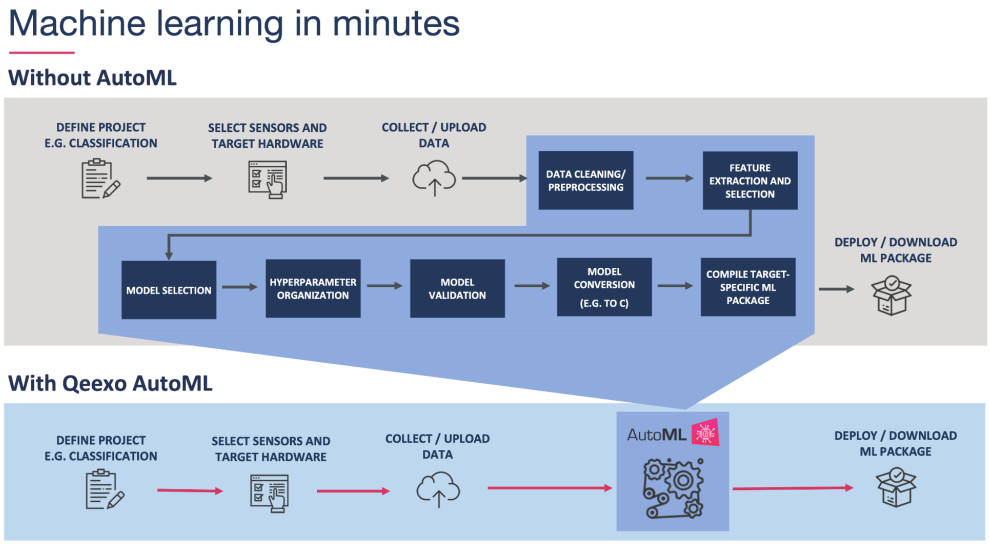
Qeexo's diverse customer base spans across fortune 500 industrials, manufacturing, food processing, mobile, and more – with Qeexo AutoML being used to build and deploy machine learning solutions on more than 400 million devices world-wide.

Use Cases and Data

Qeexo AutoML can be used with any time-series sensor data. The tool specializes in helping users quickly build models from Motion, Environmental, Electrical, and Acoustic Sensors. The no code platform can be used for any use case the user can dream up; recently our customers and partners have been focused on industrial applications like condition monitoring, predictive analytics and maintenance, anomaly detection, and product quality inspection. Qeexo also has a long history of experience with human activity recognition machine learning models going back to our founding 10 years ago. Qeexo AutoML can support any time-series sensor data imported through a CSV format. However, one of the real strengths of the platform is for integrated hardware platforms, Qeexo AutoML can automatically generate a data collection binary handling sensor configuration and data transfer to flash to your specified hardware. This eliminates the need to take up valuable embedded software engineering time building data collection applications.

Process

Qeexo AutoML can help accelerate the entire process from the start of collecting data to creating a production-ready embedded binary with optimized ML model with our easy to use platform.



Productivity

In studies the Qeexo team has done on a number of tinyML projects, we have seen tremendous gains in productivity on a number of steps of the machine learning process, often reducing from hours to minutes. The most important metric is how long it takes to deploy the final product. And for one key project the initial model deployed was done manually, requiring 4 machine learning engineers working nearly 2 months. However, the challenge was the need to deploy similar models to many other devices, and that number of ML engineers and time would not scale. With Qeexo AutoML, we were able to reduce that time to 2 field engineers, not fully trained ML engineers, and only one week to produce the production model. That type of productivity improvement enables whole new applications.

Summary

Qeexo AutoML is here to help you quickly and easily make your sensors intelligent. Learn more about Qeexo AutoML from our Demo and Concept Videos: https://qeexo.com/video/ and other resources in our Help Center: https://qeexo.com/helpcenter/. Jump right in and signup with a free trial account at https://automl.qeexo.com/

Nota AI / NetsPresso

Woneui Hong contact@nota.ai

Introduction

NetsPresso is a proprietary hardware-aware AI optimization platform which automates the development process of lightweight AI models. NetsPresso significantly reduces time and resources required to develop an AI model and optimizes it for the target device.

Use Cases and Data

Types of Data - 2D Images
Scope of support

- Task - Supervised classification and object detection tasks with neural networks
- Supported output model types - TensorRT Engine, TensorFlow Lite, OpenVINO runtime
- Supported HW - NVIDIA Jetson series (Nano, TX2, NX Xavier, AGX Xavier), Raspberry Pi series (3B+, 4B), Intel Xeon server

Special Use-Cases or Verticals - AI solutions for Driver Monitoring System and Intelligent Transportation Systems

Process

NetsPresso is a suite of services including Model Searcher, Model Compressor, and Model Launcher. It covers the overall pipeline from model training to packaging for a target device. Model Searcher covers model training and automatically searches optimized model architectures aware of the inference results on each device. Given a trained model, Model Compressor, a ready-to-use toolkit returns a compressed model within a few minutes. In the final stage, Model Launcher converts and packages the model for deployment. The three services can be used independently or in succession.

Productivity

Models for tinyML must **meet performance requirements like latency and memory footprint on the target hardware**. NetsPresso enables developers to build, optimize and deploy up to 33x lighter AI models on a RZ/V2M microprocessor of Renesas Electronics, which delivers > 2.6x faster AI inference with less power consumption. See this page for more benchmarks.

Summary

NetsPresso is a hardware-aware AI optimization platform for lightweight AI model development. NetsPresso builds, searches, compresses, and accelerates models based on the input dataset or model, and tests their performance on an actual device. An optimized model that **meets the given performance and hardware requirements can be developed in weeks, instead of months.** Visit Nota AI or NetsPresso for more information.

OmniML

Di Wu, allwu@omniml.ai

Introduction

OmniML's technology focuses on hardware-aware model compression before training, which unlocks massive potential for performance and efficiency gain. Behind OmniML, there has been years of research, innovation, and proofs of concept — not to mention founders who are accomplished in researching and engineering new AI technologies and techniques.

Use Cases and Data

OmniML software supports all ML models for deep learning such Computer Vision, Natural Language Processing, and many others. It supports time-series, camera, and radar/lidar data. OmniML supports all kinds of target hardware provided there is an existing compilation flow to deploy PyTorch models. OmniML is focusing on ADAS, autonomous driving, security camera verticals now, while also exploring early opportunities in robotics, industry automation, smart appliances, and all other edge AI verticals.

Process

OmniML's product core, Omnimizer, is integrated into the customer existing MLOps and focus on the model design, training, and re-training processes. Omnimizer exposes a set of APIs that automatically transforms an existing PyTorch model into a dynamic model format that allows for model pruning, neural architecture search and other optimizations. This dynamic model can then be customized for different hardware constraints without the need to retraining. At the end of the process, Omnimizer produces an optimized PyTorch model that can be used for downstream processes such as quantization, compilation, and deployment.

Productivity

One fundamental problem of the current TinyML development process is the manual iteration between ML engineers and deployment engineers. This often results in a long development cycle that involves multiple training and long time to market. OmniML simplifies this process by empowering MLEs to train 'deployable' from the start and exposes model tuning for deployment to the familiar program interface in PyTorch.

Summary

OmniML builds software to help customers design and train optimal models tailored to target hardware platforms that enable 20% to 5x faster models with higher accuracy and a 10x reduction in engineering and deployment efforts.

Conclusion

In this document you have seen a quick overview of 11 different tools in the tinyML Auto ML market. In the dynamic tinyML market these tools are helping power the explosive growth and new applications. We encourage you to learn more about each of the tools and stay in touch with the tinyML Foundation.

www.tinyML.org