

A discussion of deep learning approaches
and their deployment on resource
constrained IoT microcontroller
platforms

Devolving Machine Learning:

from cloud to connected end-
node

Assignment 1

Adv. Industry 4.0

Jo Edwards
2117195@STUDENT.UWTSD.AC.UK

Summary

The purpose of the report was to evaluate the changing landscape in the application of Industry 4.0 key enabling technologies, namely Machine Learning (ML) and its subfield of deep learning. The devolution of Artificial Intelligence analytics from a high level, such as business servers or cloud-based servers is an important and necessary mosaic for consideration. Several factors have been considered both in overview and by discussion of use-case studies presented in research papers.

ML frameworks which simplify the development, risks and complexity are becoming more sophisticated and optimised for deployment on the architectures most desirable for ubiquitous deployments in IoT and Industrial IoT fleets. A focus of this report is the TensorFlow framework and subjugate TensorFlow Lite model for the development of ML deep learning models tailored for fast inference execution and low impact on constrained microcontroller resource, not least the impact on energy consumption for battery powered or energy harvesting devices.

Table of Contents

Summary	1
Introduction	3
Machine Learning.....	3
Practical implementation of NN processing on a resource limited microcontroller	3
Neural Network (NN) enabled IoT applications.....	4
Cybersecurity of IoT networks	4
Case studies	4
Conclusions	6
Reference list	7

Introduction

ML leverages data and algorithms to mimic the way humans learn, to gradually improve accuracy. Over the last few decades, the availability of very large storage and computational assets have enabled innovative products and services to be deployed within the cloud and interact with remote connected devices. Libraries and frameworks such as PyWorks (Nejsun, __PYLOTS__, 2018) and TensorFlow (TensorFlow, 2019) are popular choices in rapid ML model development. Such frameworks require significant processing resource and as such remain largely deployed where such resource is readily and reliably available including desktop PCs, cloud-based servers, and task specific hardware accelerator ASICs such as intel's Movidius neural network compute device ("Intel® Neural Compute Stick 2").

More recently, the Internet of Things (IoT) have become ubiquitous within not only industrial settings, but also in our personal everyday lives. It is naturally desirable to bestow the IoT devices with some degree of ML capability (Merenda et al. 2020). However, IoT devices are normally based on microcontrollers with very limited memory footprint and relatively low processing performance.

In this report, strategies and frameworks are considered in respect of implementing ML algorithms on limited resource IoT microcontrollers, and a discussion of several examples in real world applications.

Machine Learning

Under the umbrella of Artificial Intelligence, the subfields of ML, deep learning and Neural Networks (NNs) generally reside, however deep learning is a subfield of NNs, which itself is a subfield of ML.

Traditional non-deep ML depends on human intervention, requiring human experts to construct feature sets normally with significant structured data to enable ML learning. Deep learning can use labelled data sets, a supervised learning strategy, but may also consume unstructured raw data, typically in image form or text. The advantage of unsupervised deep learning is the potential to eliminate human intervention and ingest very large data sets. This allows one to think of deep learning as a "scalable ML" (Fridman).

The "deep" in deep learning merely relates to the number of layers in a NN. a NN with three layers is just a basic NN. In considering the number of layers, the input and output layers are counted.

Practical implementation of NN processing on a resource limited microcontroller

While several ML frameworks are commercially available to enable NN processing on constrained IoT devices, TensorFlow Lite has been forefront in the development of solutions. TensorFlow Lite provides a mobile library for deploying models on connected microcontrollers and similar edge devices.

In use, a 'model' is generated, or trained model may already have been created. In a model, the inputs and outputs of the model are dynamically assigned by a ML algorithm. Data training is used to train the model to determine the most optimal parameters. The network field is subsequently mapped to the desired output based upon the iteratively trained model. If a new model is to be created, cloud-based resources such as the associated Colab Jupyter Notebook ("Google Colaboratory") provides

the means the designing and training the model in TensorFlow and brings it to production using TensorFlow Lite, in a format suitable for the target resource constrained device.

Models are readily available pretrained for common tasks, to deploy from repositories known as “Zoos” or in the case of TensorFlow, the “TensorFlow model garden” (TensorFlow).

Neural Network (NN) enabled IoT applications

Cybersecurity of IoT networks

Fleets of low-cost resource constrained IoT devices based on single board microcontroller units ranging from Arduino and STM32 up to single board portable computers such as Raspberry Pi and intel's Jetson are commonly deployed. Serious and expansive security threats both exist and are increasing exponentially in scale (Shalaginov and Azad, 2021) as the lack of standards and new technology landscapes all conspire to leave IoT with poor protection and open to novel attack.

Shalaginov and Azad, (2021, p7) state “an attacker can launch malicious activities against any components of this infrastructure: Attack 1 against IoT nodes, Attack 2 against IoT gateway, and Attack 3 against the company itself. From the perspective of cybersecurity measures, Attack 3 can be mitigated by using industrial solutions like AV/IDS, Attack 2 can be mitigated by similar solutions on the lower scale for IoT gateways, while IoT gateways do not have a standard for such measures against Attack 3.”

A novel solution proposed is founded on placing unnecessary communications off-chip by way of parameter transfer over MQTT, those transfers allowing for the training the distal NN (in the Fog or Cloud) to be adaptive to previously unseen attacks (Shalaginov et al., 2019), based on forensic data sourced from the IoT devices themselves.

Novel approaches are continually turning to multi-level (Cloud, Fog, Gateway and end-node) NN enabled strategies to defeat attacks by leveraging the small footprint, low computational load and low power communication requirements of NN model deployments by frameworks such as TensorFlow Lite.

Case studies

Vehicle Battery charge estimation

The number of embedded devices in electric vehicles (EVs) has increased significantly in recent years. An important feature of an EV is the state of charge (SoC) of the individual cells which make up the vehicle battery pack. A comparative study of two NN models was undertaken to determine the accuracy of each strategy and the relative deployment burden placed on a 32-bit constrained device, an STM32Cube.AI. (Mazzi et al., 2022).

A 1D convolutional NN (CNN) and a GRU recurrent NN (GRU_RNN) Where implemented using TensorFlow Lite. Experimental results demonstrated the superiority of the CNN in accuracy, memory utilisation, both RAM and flash, over the GRU-RNN implementation in the determination off battery cell SoC.

A similar approach using TensorFlow Lite and the STM32Cube.AI was evaluated (Crocioni et al., 2020) for SOC measurement, again the performance of the Artificial NN (ANN) deployed on the STM device was benchmarked against the TensorFlow Lite for Microcontrollers (TFLM) for the analysis and validation of the non-quantized and quantized models. STM32 implementation results outperformed TFLM ones in terms of average inference time, RAM size and Flash size.

While not strictly connected end-nodes, the SOC measurements studies via NN models on constrained microcontrollers do underline the usefulness of more accurate data harvested from small microcontroller platforms.

Human Activity Recognition

Human activity is the focus of intense research interest. Categorising human activity using wearable devices capable of analysing and communicating descriptive data to cloud-based services. Random Forests (RF), a classical ML technique and deep learning Convolutional NNs (CNN), both custom and state-of-the-art, are compared (Elsts and McConville, 2021)

The PAMAP2 dataset (mldta.com, n.d.) contains data of multiple physical activities performed by 9 subjects wearing 3 inertial measurement units (IMU) was used as source data to evaluate the ML efficacy on several different Cortex-M core microcontrollers of the RF and custom and standard CNN models.

It was found that the custom CNN, being much smaller and more efficient, outperformed the standard CNN in inference speed and quantization burden (32b to 8b for model input dataset compatibility), the faster execution speed also reducing energy consumption significantly. However, the RF classifier outperformed the custom CNN classifier, requiring significantly less human input to design.

The results do cast a doubt that deep learning inference based NNs are always the optimal solution for every application.

Autonomous vehicles

Until recently unmanned aerial vehicles (UAVs) were uncommon. That has changed considerably as the related technology platforms proliferate and benefit from much lower cost of ownership. Real-time risk and opportunity assessment has significant advantage over the consequences of longer than desirable communication latency with a remote off device decision-making resource. An OpenMV Cam H7 MCU (OpenMV, 2018), compatible with TFLM framework models, was integrated with a commercially available drone (Raza et al., 2021).

The task of the automated drone was to detect humans wearing PPE masks, a convolutional NN (CNN) trained to detect face masks used the MobileNet V2 architecture (Sandler et al., 2018) using a publicly available dataset (www.kaggle.com, n.d.).

Evaluation of non-quantized against quantized models showed the dynamically quantized network model reduced the inference time by approx. 50% and the RAM usage by 60% whilst maintaining a post training validation accuracy of 97%. The benefits could be attributed to the significantly faster processing of smaller code and memory efficiencies of 8bit over 32bit data processing on the platform used.

Conclusions

The case studies highlighted the high degree of efficacy of TensorFlow lite, and related TensorFlow Lite for Microcontrollers (TFLM) as a tool for benchmarking performances, bestowed on custom deep learning strategies for IoT autonomous task optimisation.

We considered

- Hardening cybersecurity by distributed threat detection through dynamically training a CNN model, allowing forensic data to be collected on attack evolution.
- Replacing standard SoC algorithms with deep learning solutions at the 'tiny' platform level, to optimise EV battery life and charge cycles.
- Reducing decision making latency in UAVs by onboarding custom CNNs, increasing the utility of the UAVs and advancing mission capabilities at low to modest costs in hardware additions.
- And finally, in the field of human activity recognition, a case study which serves to make a developer pause and consider, is deep learning always the panacea to very small platform opportunity?
- Or depending on task, is tradition ML more appropriate in accuracy and execution time and costs?

Reference list

A. Shalaginov, I. Kotsiuba and Iqbal, A. (). Cybercrime Investigations in the Era of Smart Applications: Way Forward Through Big Data. In: *2019 IEEE International Conference on Big Data (Big Data)*. pp.4309–4314. doi:10.1109/BigData47090.2019.9006596.

Blacker, P., Christopher Paul Bridges and Hadfield, S. (2020). Diagonal memory optimisation for machine learning on micro-controllers. *CoRR*, [online] abs/2010.01668. Available at: <https://arxiv.org/abs/2010.01668>.

colab.research.google.com. (2022). *Google Colaboratory*. [online] Available at: <https://colab.research.google.com/notebooks/welcome.ipynb> [Accessed 2022].

Crocioni, G., Pau, D., Delorme, J. and Gruosso, G. (2020). LiIon Batteries Parameter Estimation With Tiny Neural Networks Embedded on Intelligent IoT Microcontrollers. *Access*, 8, pp.122135–122146. doi:10.1109/ACCESS.2020.3007046.

Elsts, A. and McConville, R. (2021). Are Microcontrollers Ready for Deep LearningBased Human Activity Recognition? *Electronics (Basel)*, 10(21), p.2640. doi:10.3390/electronics10212640.

Fridman, L. (2019). *MIT Deep Learning Basics: Introduction and Overview*. YouTube. Available at: <https://www.youtube.com/watch?v=O5xeyoRL95U>.

Intel. (2018). *Intel® Neural Compute Stick 2*. [online] Available at: <https://www.intel.com/content/www/us/en/developer/tools/neural-compute-stick/overview.html> [Accessed 11 Aug. 2022].

KOTTARATHIL, P. (2020). *Face Mask Lite Dataset*. [online] www.kaggle.com. Available at: <https://www.kaggle.com/datasets/prasoonkottarathil/face-mask-lite-dataset> [Accessed 11 Aug. 2022].

Mazzi, Y., Sassi, B., Gaga, A. and Errahimi, F. (2022). State of charge estimation of an electric vehicle's battery using tiny neural network embedded on small microcontroller units. *International journal of energy research*, 46(6), pp.8102–8119. doi:10.1002/er.7713.

Merenda, M., Porcaro, C. and Iero, D. (2020). Edge Machine Learning for AIEnabled IoT

Devices: A Review. *Sensors (Basel)*, 20(9), p.2533. doi:10.3390/s20092533.

Nejsum, __PYLOTS__, R. (2018). *pyworks: An Task framework*. [online] PyPI. Available at: <https://pypi.org/project/pyworks/> [Accessed 12 Aug. 2022].

OpenMV. (2018). *OpenMV Cam H7*. [online] Available at: <https://openmv.io/products/openmv-cam-h7>.

Raza, W., Osman, A., Ferrini, F. and Natale, F.D. (2021). EnergyEfficient Inference on the Edge Exploiting TinyML Capabilities for UAVs. *Drones (Basel)*, 5(4), p.127. doi:10.3390/drones5040127.

Reiss, A. (2012). *PAMAP2 Physical Activity Monitoring | Computer Science Dataset*. [online] mldta.com. Available at: <https://mldta.com/dataset/pamap2-physical-activity-monitoring/> [Accessed 12 Aug. 2022].

Rua, S., A, Z.S., Redondo, A., OrozcoDuque, A., V, R.J. and Bustamante, J. (2012). Machine learning algorithms for real time arrhythmias detection in portable cardiac devices: microcontroller implementation and comparative analysis. In: *STSIVA*. IEEE, pp.50–55. doi:10.1109/STSIVA.2012.6340556.

Saha, S.S., Sandha, S. and Srivastava, M. (2022). *Machine Learning for MicrocontrollerClass HardwareA Review*. doi:10.48550/arXiv.2205.14550.

Sakr, F., Bellotti, F., Berta, R. and Gloria, D. (2020). Machine Learning on Mainstream Microcontrollers. *Sensors (Basel)*, 20(9), p.2638. doi:10.3390/s20092638.

Sandler, M., Howard, A., Zhu, M., A. Zhmoginov and Chen, L.C. (). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp.4510–4520. doi:10.1109/CVPR.2018.00474.

Shalaginov, A. and Azad, M.A. (2021). Securing ResourceConstrained IoT Nodes: Towards Intelligent MicrocontrollerBased Attack Detection in Distributed Smart Applications. *Future internet*, 13(11), p.272. doi:10.3390/fi13110272.

Shalaginov, A., Semeniuta, O. and Alazab, M. (2019). MEML: Resource-aware MQTT-Based machine learning for network attacks detection on IoT edge devices. In: *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*

Companion. [online] Auckland, New Zealand: Association for Computing Machinery, pp.123–128. doi:10.1145/3368235.3368876.

tensorflow (2019). *tensorflow/models*. [online] GitHub. Available at: <https://github.com/tensorflow/models>.

TensorFlow. (2019). *TensorFlow Core*. [online] Available at: <https://www.tensorflow.org/overview>.

TensorFlow. (2022). *TensorFlow Lite guide*. [online] Available at: <https://www.tensorflow.org/lite/guide> [Accessed 10 Aug. 2022].