iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: XI Month of publication: November 2021

DOI: https://doi.org/10.22214/ijraset.2021.39067

Embedded Machine Learning and Embedded Systems in the Industry.

Pratic Chakraborty
A Student of B.Tech in Applied Electronics an Instrumentation Engineering. Currently Studying at 3rd year.
Academy Of Technology (Affiliated to MAKAUT University)

Abstract: Machine learning is the buzz word right now. With the machine learning algorithms one can make a computer differentiate between a human and a cow. Can detect objects, can predict different parameters and can process our native languages. But all these algorithms require a fair amount of processing power in order to be trained and fitted as a model. Thankfully, with the current improvement in technology, processing power of computers have significantly increased. But there is a limitation in power consumption and deployability of a server computer. This is where "tinyML" helps the industry out. Machine Learning has never been so easy to access before!

I. OBJECTIVE

The main objective of using machine learning algorithms on embedded devices are to increase the number of use cases as they consume less power and are very durable and reliable.

II. INTRODUCTION

According to the Moore's law "The number of transistors double every year, in a processor". Unfortunately the statement was prooven quiet wrong as modern desktops or laptops contain trillions of transistors now a days. With this abruct rise in number of transistors, clock speed of the processors increased, I/O capability and interfacing options increased, as well as the number of cores in a processor increased. Resulting in thousands of times faster computational capabilities. But with this increased power, comes increased power consumption, greater heat generation and the setups usually get pretty costly to be used in every application. On the other hand however modern day to day life runs on AI and machine learning. But as mentioned earlier, servers can be very costly and hard to maintain. As a solution to this problem a new trend emmerged, called "tinyML". In which machine learning models are run smaller and more scalable hardware like micro controllers, SoCs and FPGAs. Although training the models are done on a server or a desktop but the models are deployed in these handy little devices. The software package used to deploy the models to these embedded platforms(systems) is called "TensorFlow lite". Which is a subset of "Tensorflow", which is used to train the models in a server or a desktop. There are many hardware accelerators, which can speed up the training process of the model and can also speedup the overall processing speed of the system.

III. WHAT IS MACHINE LEARNING? (IN A BRIEF)

Machine Learning is the process by which a computer figures out the relation between the data and the given label to that data. For example, in typical programming we feed data and the set of rules to the computer and the computer gives us the output. But in machine learning we feed the computer the data and the correct output regarding the given data. Then the machine learns the pattern and gives us set of rules as an output.

By using machine learning machines are being capable of distinguishing between objects, animals (Computer vision problems) and are being able to predict some data based on some training data.

There are many different algorithms to make the computers learn the pattern between the data and the labels like clustering, classification, regression etc. And with rise in computational power and amount of available data a newer stream of machine learning came to focus, which is known as Deep Learning. Which uses layers and nurons, to do the exact same thing like machine learning. But a deep learning model can be more accurate and can deal with more complex problems than traditional machine learning algorithms.

IV. WHY USE A MICRO CONTROLLER (OR A SOC) INSTEAD OF A DESKTOP OR LAPTOP?

This question can be answered in many ways. First one being, power consumption. A micro controlller draws a few miliWatts of power where a Desktop need hundreds of watts to operate. The second one being the fact that micro controllers are lot cheaper and easy to produce than a laptop or a desktop. A micro controlller can handel much more harsh conditions than a desktop which expands the use cases even further. And last but not the least is the form factor. A full micro-controlller board can fit on somebody’s finger tip but a laptop or desktop cant be placed there.

V. HOW THE HARDWARE AFFECTS THE ABILITY TO RUN MACHINE LEARNING MODELS

Lets discuss this part by part:-

1) *The Processor:* The processor is the brain of a ccomputer or an embedded system. Faster the processor is, faster the calculations can be performed. In the application of tinyML or embedded machine learning the most suitable processing cores are ARM processors. There are plenty of types of arm processors, starting from ARM cortex M0 and all the way up to ARM cortex A57 processors. With better architecture, efficiency and the speed increases. This can better be understood from figure 1.1.

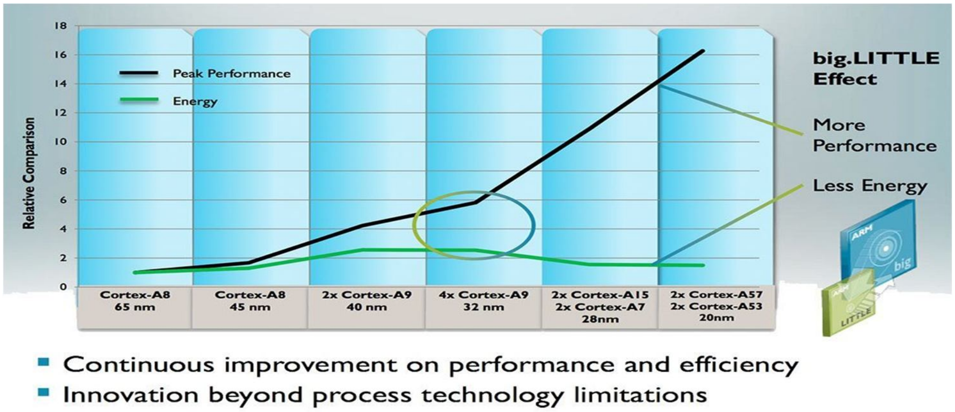

Figure 1.1

2) *RAM:* RAM is required as a space to store the variables, larger the space the more variables it can fit. So the batch size of the algorithm can be larger hence performance can be improved.
3) *Secondary Storage:* The secondary storage is the place where the program and data are stored. It can be in the form of ROM, Flash or EMMC storage.

Now, the above mentioned points are true for both a micro controller and a single board computer.

For the single board computers things get a little different as they can work as a full computer. Hence training a model on them might be possible but, we have to keep in ming that the **lithography** of a single board computer’s CPU is way smaller than a Desktop or a laptop. So clock speeds are kept low in order to reduce heat generation and power consumption. So the process might be significantly slower.

VI. WHAT IS TENSORFLOW AND TENSORFLOW LITE

Tensorflow is an open source library made for training and creating **Deep Learning** models. It is mainly programmed in python but C++ and Java are also supported by the Tensorflow. It was developed by Google. Working with Tensorflow can be fairly complicated as the tools are very resource hungry, and can lead to a system crash the system is not capable enough. However single board computers like the Raspberry pi 4 and NVIDIA Jetson NANO can run Tensorflow on them. Tensorflow lite on the other hand is a subset of Tensorflow. Which means it has much less built in functions and can perform much less tasks. This is mainly intended for edge devices like Micro-controlllers. With much less built in functions, tensorflow lite is very small in size which can fit as a library in boards like Arduino Nano 33 BLE Sense. Tensorflow lite uses a function callled “quantization” to convert all unncecessary floating point numbers into 8 bit or 16 bit unsigned integers. Which drastically reduces the RAM usage.

1) Now a days there are plenty of ARM based micro controller boards like the Raspberry pi Pico(M0+), ESP32(Espressif, M0+), Arduino nano 33 ble Sense(ARM cortex M4) which have made the process of deploying embedded machinelearning models very easy and efficient.
2) On the other hand there are several single board computers out there which can run OpenCV and Tensorflow on them. Like Raspberry Pi 4, NVIDIA Jetson NANO etc.

VII. HARDWARE ACCELERATORS

The hardware accelerators are very specialized kind of hardware which are meant to do only some specific tasks, but they do those faster than any other device.

There are two types of hardware accelerators used to speed up learning process.

1) *GPU:* The word GPU stands for **G***raphical* **P***rocessing* **U***nit*. These have a massively parallelize able architecture consisting of blocks and grids. Each block can process 1024 threads and a grid can have 65535 blocks inside it. There are many GPUs aavilable in the market but the NVIDIA ones are programmable and can be used as a hardware accelerator. For programming the GPU a special programming language is used called CUDA (Compute Unified Device Architecture). Although NVIDIA GPUs can also be programmed in python by using different CUDA based APIs like “pyCUDA”. Tensorflow can use this CUDA toolkit to accelerate the training process. Below is a picture of nvidia GTX 1650 GPU (Figure 1.2).

Figure 1.2

2) *TPU:* TPU or **T***ensor* **P***rocessing* **U***nit(s)* are a more specialised kind of hardware made specifically for processing **tensors** and accelerating A.I. applications. Arrays declared and used in tensorflow are known as tensors. The tensor processing units come in all sorts of interfacing options starting from USB plug and play to a faster and more powerful PCI connectivity options. Intel Movidius V1 and V2 are probably most widely used TPUs now a days as other kinds of TPUs are only available for consumers through cloud providers.

Below is a picture of Intel Movidius (Figure 1.3):-

Figure 1.3

Although hardware accelerators comein really handy when it comes to processing images and training datasets with large matrices or dimentions. But they are very costly for some low budget applications and are not always feasible to use like for relatively smaller datasets. A CPU would perform better than a GPU for smaller datasets due to data transfering latency. The graph below will give a better intuition. (* The Graph was plotted using Matplotlib and python 3.8)

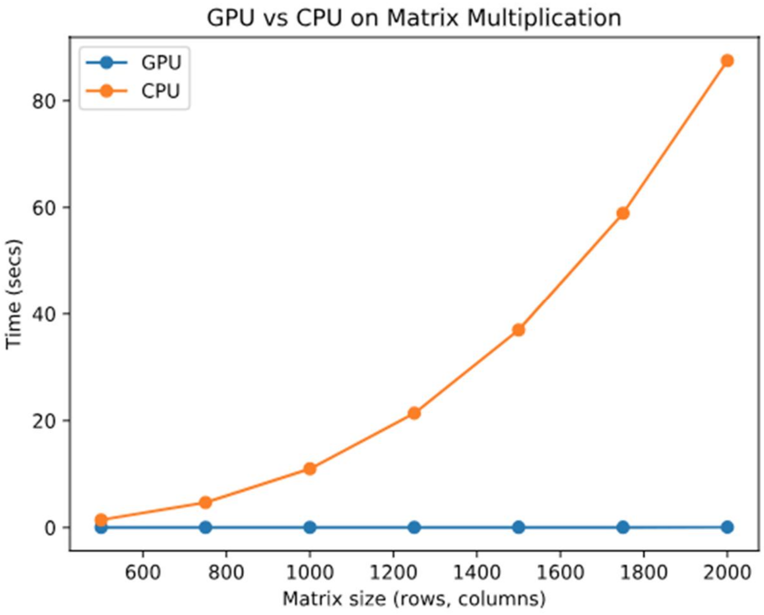

Figure 1.4

Some Single Boards Computers like the NVIDIA Jetson Nano come with an ARM cortex A 57 quad core CPU and a GPU of “Volta” architecture, which has 128 CUDA capable cores. That kind of specification allows us to implement hardware acceleration in embedded systems.

VIII. CONCLUSION

A. The main point of using embedded systems and micro-controllers to deploy AI models are that these devices are very power efficient, very durable and have very small form factor which makes it even easier to deploy in use cases like wearables.

B. Micro controllers and Single Board Computers (SBC) are very cheap as compared to a full sized desktop or a server. And they are easy to maintain too.

C. Embedded systems are very realiable and given the point that some embedded systems like the NVIDIA Jetson Nano have built in GPU(hardware accelerator) on them makes these systems even more efficient for AI and machinelearning.

REFERENCES

[1] TinyML: MachineLearning with Tensorflow Lite on Arduino and Ultra-low power micro controllers, By – Pete Warden & Daniel Situnayake, Published by – O’Reilly Media

[2] NVIDIA CUDA documentation :- https://docs.nvidia.com/cuda

[3] Tensorflow Documentation:- https://www.tensorflow.org/api_docs