Machine Learning in Embedded System

Vijay Natarajan

Program Manager

Embedded Product Design

TATA ELXSI Ltd

Bengaluru, India
vijayn@tataelxsi.co.in

***Abstract*—this paper discusses the general opportunities available using the technologies options to port any Machine Learning in Embedded System.**

Keywords—Machine Learning, Embedded System Algorithms, Advanced Technologies in semiconductor

I. INTRODUCTION

Why Should Machine Learning Algorithm Reside in Cloud?

If Machine Learning has to touch human lives, it just can't reside in Cloud; it has to come down to Earth and live in Embedded System. Recent advances in real time systems demand complex applications requiring intelligent behaviour, while Machine Learning is heading towards more realistic domains that require real –time responses.

Window of opportunity exists at the intersection of Machine Learning and Embedded System, which demands hard real time with deadlines to produce response. Failure will be catastrophic for Artificial Intelligence in health care & self-driving vehicles.

Extracting useful information from zeta bytes of sensor data is a "Cognitive Overload" for Humans however, Machines that "learn", can drive actionable insights from sensor data.

Traditional Machine Learning Algorithm residing in Cloud for computation suffers from Privacy, Latency and Communication bandwidth concerns. Hence Machine Learning Algorithms in Embedded System nearer to Sensors for local processing makes a lot of sense to overcome the disadvantages of Cloud Architecture. However Machine Learning in Embedded system poses some serious challenges to System design. We are aware that Embedded Systems are "Bounded Rationality Systems" which means that, they are stringently constrained systems while considering the Cost, Energy Consumption and Size (Real estate).

Challenges posed by Machine Learning in embedded system can be addressed at various levels of Hardware design ranging from Architecture, Hardware friendly Algorithms, use of Mixed Signal circuits and adopting Advanced Technologies. Thus reducing Energy Consumption, size and cost of porting Machine Learning Algorithm in Embedded Systems.

II. CHALLENGES OF MACHINE LEARNING IN EMBEDDED SYSTEMS

Machine Learning is known to be compute intensive for learning complex features. Regular CPUs found in Embedded System just aren't well suited to the demands of Machine Learning. Trying to force fit them makes it slower to deliver the services and their high energy computation drains the battery in wearable devices.

A. Energy Consumption

Programmability increases data movement and memory storage requirement increases Energy consumption

B. Latency and throughput

Computational capacity and latency increases with dimensionality of data

C. Cost

Cost is governed by the amount of memory storage required in the system

D. Accuracy

Accuracy requires large datasets, at times large data can even beat the best Algorithms

III. APPLICATIONS OF MACHINE LEARNING IN EMBEDDED SYSTEM

Google's Mobile AI frame work, TensorFlow Lite is designed for embedded platform and is readily available for Android, Raspberry and iOS. Lean Mobile Apps reduces the models foot print with the use of optimizing technique. For example, iPhone's face detection feature, Google's instant visual translation, speech recognition and synthesis in embedded products and Machine Vision for object detection in video surveillance cameras.

IV. OPPORTUNITIES FOR MACHINE LEARNING IN EMBEDDED SYSTEM.

In the recent months we have heard about specialised silicon used for Machine Learning in mobile devices. Apple's new iPhones have their 'neural engine', Huawei's Mate 10 with 'neural processing unit' and companies that manufacture and design chips (like Qualcomm and ARM) are gearing up to supply AI optimised hardware to the rest of Industry.

Opportunities come with challenges. Research in AI has shown various options for porting Machine Learning algorithms on embedded system that includes optimization in

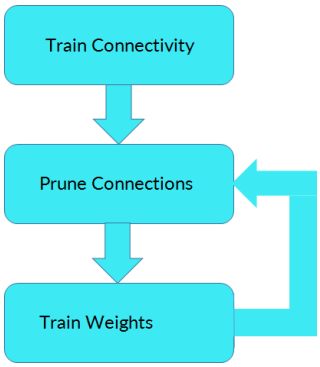
- Algorithms
- Advanced Technologies
- Mixed Signal Circuits

A. *Opportunties in Algorithms*

Modifying Machine Learning algorithms to make them more hardware-friendly, while maintaining accuracy the focus is on reducing computation, data movement and storage requirements

1) Opportunities in learning connections

a) Learning connections via normal network training: Learning connections via normal network training, we are not only learning the final values of the weights, rather one learns which connections are important. If the pruned network is used without retraining then the accuracy is significantly impacted. As shown in Fig 1.



- Step1: Learning which connections are important
- Step2: Prune the low-weight connections. All connections with weights below a threshold are removed from the network
- Step3: Retrains the Network to learn final

Fig 1. Three step Training pipe line

b) Regularisation: Choosing the correct regularization impacts the performance of pruning and retraining. L1 regularization penalizes non-zero parameters resulting in more parameters near zero. This gives better accuracy after pruning, but before retraining. However, the remaining connections are not as good as with L2 regularization, resulting in lower accuracy after retraining.

c) Dropout ratio adjustment: Each parameter is probabilistically dropped during training, but will come back during inference

d) Interactive pruning: Boosts pruning rate from 5 to 9 on AlexNet compared with single-step aggressive pruning

e) Pruning Neurons: Dead neurons will be automatically removed during retraining

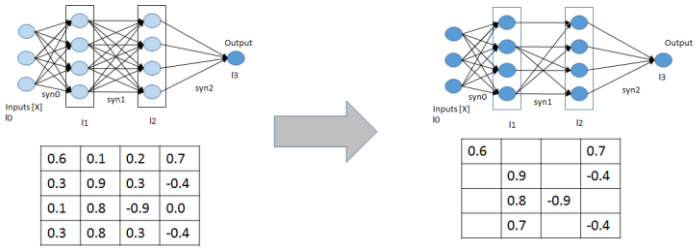
2) Compression in Algorithms

Compression – preventing excessive Data movement and Reduce Memory storage saves Cost and Energy Three Techniques for compression of Neural Network

- Sparsify
- Shrink

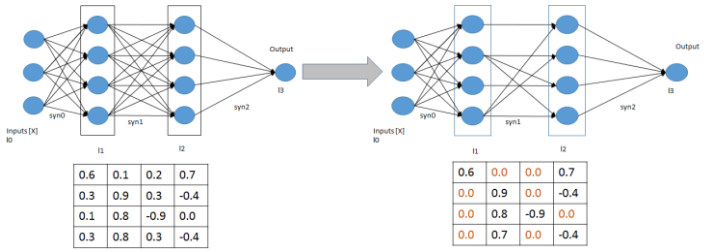
a) Sparsify the Matrix

Storing only Non-zeroes significantly reduces storage space ~5x Compression achieved in Model



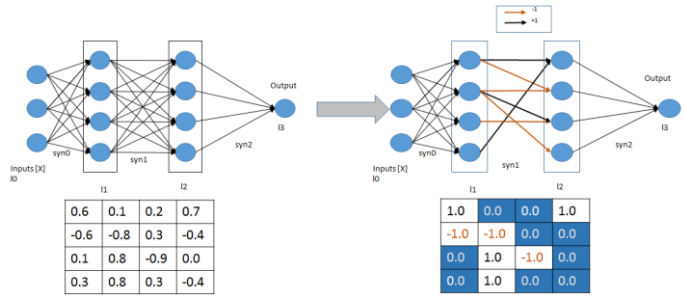
b) Shrink the Matrix

Minimal architecture design of Neural Network, selecting appropriate filters reduces compression by ~3x



3) Quantise the matrix

The Reduce precision for energy and storage savings by reducing to 8–bit Integer from 32bit or 64 bit as present in CPUs & GPUs saves Energy by 2.56 x and increase in throughput by 2.24 x. This can be achieved by quantisation.



B. *Opportunities in Advacnce Technologies*

1) Advanced memory technologies such as embedded DRAM and Hyper Memory Cube to reduce energy access cost of the weights in DNN

2) Research is exploring integrating the multiplication directly into advanced non-volatile memories by using them as resistive elements

3) ReRAM(Resistive RAM) store and process on the same chip, is used to compute the product of a 3-bit input and 4-bit weight.

C. *Opportunities in Mixed Circuits*

1) Mixed-signal circuits to reduce the computation cost of the MAC (Multiplications and Accumulation) in Analog domain

2) 3-bits and 6-bits are sufficient to represent the weights and input vectors, reduces the number of ADC conversions by 21 and sensor bandwidth by 96%

3) Matrix multiplication integrated into the ADC where the MSB of the multiplications for ADA boost classification are performed using switched capacitors in an 8-bit successive approximation format, despite ADC/DAC conversion overhead

V. Conclusion

Machine Learning with many promising applications provides an opportunity for innovation and creativity at various levels in Hardware design and Algorithms.

Algorithms are motivated by how learning works in mammalian brain that operates by learning the connections that are important, pruning the unimportant connections and then retraining the remaining sparse network. This improves energy efficiency and storage of neural networks without affecting accuracy by finding the right connections.

AI Framework for developers from Google like, TensorFlow Lite, has already standardized some experiences on mobile devices, and its own Android-wide APIs to "tap into silicon-specific accelerators."

Specialized Machine Learning Hardware for Embedded Systems means in theory, better performance and battery life, reinforcing user's privacy and security.

We can realize interesting Intelligent IoT applications if Machine Learning resides near sensors and not in Cloud.

.

References

.

[1] Vivienne Sze, Yu-Hsin Chen, Joel Emer, Amr Suleiman, Zhengdong Zhang, "Hardware for Machine Learning: Challenges and Opportunities,"

[2] Wenlin Chen,James T. Wilson,Stephen Tyree,Kilian Q. Weinberger,Yixin Chen "Compressing Neural Networks with the Hashing Trick"

[3] Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting.The Journal of Machine Learning Research, 15(1):