Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN: 2581 - 611X)
VOLUME:8, ISSUE:2, AUGUST 2025, PAPER NUMBER:51

Design And Implementation Of Approximate Multiplier For CNN Applications

Bhuvaneshwaran S[1], Gukan Prasath K [2], Thamilthennarasu M [3], Vishal Kumar M G[4]

[1,2,3,4] Students
Dept. of Computer Science Engineering and Information Technology
Bannari Amman Institute of Technology
Sathyamangalam, Erode
bhuvaneshwaran.cs21@bitsathy.ac.in

Abstract: *Recently, Convolutional Neural Networks (CNNs) have greatly impacted numerous fields including image identification, voice recognition, and self-driving systems. However, this kind of convolutional networks have a relatively high computational complexity and often times need high performance hardware and thus there is high power consumption. This has lead to the rising interest in approximate computing, as an effective approach to addressing this growing demand. Through approximation, it is possible to establish reasonably small error margins for certain computations, which can significantly enhance the efficiency of the approximate computing system while still providing sufficient performance. This paper presents the architectural enhancement and the software approach of an approximate multiplier tailored for CNNs. This new multiplier employs the following software methods; bit-width reduction, precision scaling, quantization, stochastic operation and controlled methods of rounding off. Altogether, these strategies improve energy efficiency, computation speed, and hardware design complexity while maintaining the CNN performance below the certain limit. In addition, we provide an extensive analysis of the applied implementation over a number of popular CNN architectures and show impressive improvements with regards to computational efficiency and energy consumption. These enhancements suggest that approximate computing can be innovative in enhancing the effectiveness of CNNs in practical use in terms of time efficiency with no significant impact on classification accuracy based on our experiments.*
Keywords: *Approximate Computing, Approximate Multiplier, Convolutional Neural Networks (CNNs), Precision Reduction, Bit-width Reduction, Energy.*

1. INTRODUCTION:

Convolutional Neural Networks (CNNs) have established themselves as one of the pillars of the current artificial intelligence. especially when it comes to such applications as image sensing, visioning, and sign pattern recognition. The structure of the architecture they apply is based on the neurons of the biological visual cortex which react certain specific stimuli; for example, edges and textures. In CNNs, this process is reflected by layers of convolution filters that perform feature extraction of hierarchal structure from the input data that can be an image or a video stream. As a In terms of their performance CNN shows remarkable abilities to capture image features and shapes together with the identification of objects regardless the circumstances and conditions. noise, lighting, and transformations. The general applicability of CNNs have been witnessed in real life problems, for example; autonomous driving, facial recognition, medical diagnosis, and video surveillance, has led to higher demand for using these models on any type of device you have. From computing centres performing large scale inference activities such as reception of task to mobile devices performing real-time facial recognition, the need for highly efficient computation has never been a greater challenge.

However, the shared power consumption and the highly computational task load of CNNs present great challenges, primarily to edge computing devices with restricted computational capabilities and energy resources.

CNNs are in some way computationally intensive especially at the time of forward propagation where most. It is noteworthy that operations are matrix multiplications between input feature maps and convolution filters. These calculations, which are usually performed in high accuracy (analogous to 32-bit or 64-bit floating point) of this type, are not only computationally expensive but also energy-intensive. While the use of deep learning algorithms requires significant computational resources and time, it is also a hardware power hog. For example, using CNNs on low power devices as well as incorporating the change in the scale and distribution of the problem into the design of the solution tablets, smartphones, drones and IoT sensors that can significantly reduce battery life to extremely long in any life cycle as there is so much multiplications to be done for each element. As the screen shots of CNN models go deeper and more, to be made in order to enhance the accuracy of structures which are complex, the energy and the computational requirements rise significantly.

As a result, approximate computing has emerged as a promising paradigm to address the computational bottlenecks in CNNs. The main concept of approximate computing is concerned with the understanding that all calculations demand exact zero tolerance, as common to most situations that demand rigid precision in some intermediate calculations can be excluded do not have a substantial effect on the final outcome of the program. In image and video processing, for example, small inaccuracies almost always escape the human eye, which is why such applications are best suited approximation. CNNs, in their operating principle, are capable of handling small errors because of their structure. summarized data at multilevels of computation and machining. It is this resilience that make it possible to introduce partially inaccurate or approximate impressions of some occurring nontechnical parameters without much impact overall model performance. Under such circumstances approximate multipliers may be helpful by themselves. Multiplication is one of the most time- and power-consuming operations in CNNs. By reducing the precision of multiplication operations, or by employing mathematical shortcuts that approximate multiplication, we can significantly reduce computational cost and power consumption. These approximations can be used selectively as in the non-critical layers of CNN where slight fluctuations in calculation may not impact the final output and hence with a large degree of change the efficiency can be achieved with little consolation offs in accuracy.

2. DESIGN OF APPROXIMATE MULTIPLIERS :

As the market for deep learning solutions stays actively growing, especially considering the Convolutional Neural Networks (CNN), the computational demand for these models has escalated dramatically. Approximate multipliers offer a persuasive solution which allows for cuts to power and area and it as permitted some inaccuracy in both of them. The underlying principle of approximate computation is based on the observation that many applications, especially in the realm of machine learning and CNNs, can tolerate a certain degree of error without a significant impact on overall performance. By strategically introducing approximation into multiplication operations, we can achieve notable enhancements in speed and efficiency.

A. Bit-Width Reduction

The simplest form of design for approximate circuits is known as bit-width reduction technique. Reducing the number of bits, in which numerical values are represented, is the idea behind this method during multiplication operations. The primary rationale behind bit-width reduction is the

realization that not all bits are equally important for every computation, especially in the context of neural networks, where certain weights and activations can be represented with fewer bits without significantly degrading the performance. Several methods were used in carrying out bit-width reduction; the methods include the truncation of the least less significant bits – LSBs and fixed – point representation. For example, moving from a 32–bit architecture to a 16-bit or even 8-bit fixed-point format will ease the arithmetic operations, time consuming and large size of network that the corresponding mathematical operations and greatly enhance the speed of the computation. This transition does not only lower down the hardware exactly to the complexity of the multiplier circuit but also optimizes the power consumption and memory bandwidth usage.

Bit width	Reduction in power consumption
4	20.51%
8	35.25%

Fig. 1. Power Consumption

B. Fixed-Point Arithmetic

Fixed-point arithmetic represents a compelling alternative to floating-point arithmetic in the design of approximate multipliers. In fixed-point representation numbers are written in the form of fixed set of digits and thus the name fixed point representation precision in terms of the number of decimal places, and provide better results using multiplication than floating-point arithmetic which demands more complex series of operation to manage the exponent and mantissa. The major benefits of using fixed-point arithmetic relate in its simplicity, and performance. Fixed-point multipliers generally need less hardware and power than the floating point types of multipliers making them ideal for embedded systems. This decrease of complexity leads to faster computation times which is important for application that need to perform in real time. Perhaps the most important area in the successful implementation of strategic management is the use of fixed-point multipliers does not present many problems, however an important point is the choice of the scale factor. The scaling factor determines the place of the binary point in the representation and affects the dynamic range as well as the precision of the numbers processed.

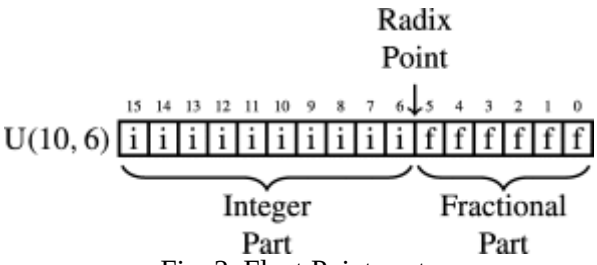


Fig. 2. Float Point parts

C. Quantization Techniques

Quantization is a very important step in the design of approximate multipliers, because it allows the discretization which is the process of converting the numerical data into a finite range of integer values. This technique is especially beneficial in CNN's because the weights and activation can be quantized hence minimizing the computational requirements while at the same time not affecting the performance in a very negative way.

- Post-Training Quantization: This approach means that the weights that are in advance need to be quantized. Transitioning from high precision for example 32-bit floating point, into a lower precision format (e.g., 8-bit). From the distribution of weights, the researchers are in a position to reduce the quantization error and for the model to still be as accurate as possible.
- Quantization-Aware Training (QAT): In QAT, improved quantization is at the same incorporated. training process. This enables the model to understand how best to reduce impact of quantization during

training, they get a quantized model that is more resistant to some added errors caused by decrease in precision.

D. Rounding Strategies

Rounding methods are the key issue of approximate multiplier design because they define how in computations the numerical values on the scale are flexible and amended. Thus, it is critical to understand that the option between rounding methods can make significant consequences to the total reliability of the multiplier results.

- Nearest Rounding: This traditional approach involves rounding the result to the nearest representable value. While simple and effective in many cases, it can introduce systematic errors if not applied judiciously.
- Stochastic Rounding: To achieve rounding, stochastic rounding adopts a measure of probability perspective choices, which will make it possible to provide better identification of values in different decisions and several. operations. Thus, as a result of rounding based on the fractional part of a number this technique can serve to reduce bias and therefore provide fairly stable results.

E. Hybrid Approaches

The advantages of each system are then used in a hybrid approach that employs a combination of the approaches. about designing approximate multipliers. It is there thus possible to obtain a single key mix, effective at integrating various strategies for the creation of the similar key mix throughout the convergence processes. they provide multifunctional motors that perfectly combine factors such as speed, power and precision.

- Combining Techniques: For instance, a hybrid design might use fixed-point representation alongside stochastic computing to minimize both hardware complexity and power consumption. By strategically applying bit-width reduction and quantization techniques, hybrid multipliers can achieve high efficiency while maintaining acceptable accuracy levels.
- Optimization of CNN Layers: Hybrid designs can also optimize specific layers within a CNN. For example, convolutional layers that are less sensitive to precision might employ approximate multipliers, while fully connected layers, which often require higher precision, can use traditional exact multipliers. This targeted application ensures that the model maintains high performance where it matters most.
- Impact on Deployment: The use of hybrid approaches enhances the feasibility of deploying CNNs in resource-constrained environments. By tailoring the multiplier design to the specific requirements of the model, developers can ensure that the resultant application operates efficiently on limited hardware.
- Future Directions: Research into hybrid approaches is ongoing, with a focus on developing new techniques that can further improve efficiency without sacrificing accuracy. As the field of deep learning continues to evolve, hybrid designs will play a crucial role in enabling the deployment of advanced models on a wide range of devices.

3. Implementation of Approximate Multipliers in CNNs:

The exploitation of approximate multipliers in Convolutional Neural Networks (CNNs) is shown to be challenging. an innovation and a major step forward particularly in enhancing performance and productivity. By strategically integrating these multipliers, researchers can decrease the necessary computational time and energy expenses. As a result of which, important calculations were maintained at a reasonable level of accuracy.

A. Layer-Specific Implementations

Several layers of CNN can require approximate multiplication in one way or another. Each layer system plays a definite role within the network, and knowledge on these roles is important in the network. optimization.

- Convolutional Layers: These layers entail many multiplications to both weights and inputs as such they are the ideal candidates for approximation approaches. The large number of that is why anything below full precision results is critical, and even minor deterioration in terms of precision can have significant impact on the overall efficiency of performance. Therefore, by using techniques like reduced bit-width and fixed point arithmetic the execution period of the LMS algorithm can be reduced. representation, which shows that with only a very small penalty in number crunching, convolutional layers can realize huge time-savings. on overall accuracy.
- Activation Layers: An activation function is important when we seek to perform non-linear transformation of data in CNN. The subsequent outputs of layers containing convolution can be handled by approximate filters before passing it to activation functions such as ReLU or sigmoid functions. This approach may optimise work, boosting computational speed, but main functionality does not have to be burdened with design of a model that is adaptive to change in the inputs.
- Pooling Layers: The pooling operations then do not contain multiplications but are far from simple since conventional. It is therefore possible to incorporate several pooling strategies such as a learned pooling or attention mechanisms for approximate multipliers. Using of these multipliers in adaptive pooling operations can enhance efficiency when it comes to procedures that concern sophisticated extraction of features, especially on models that contexts characterised by high variability in terms of inputs.

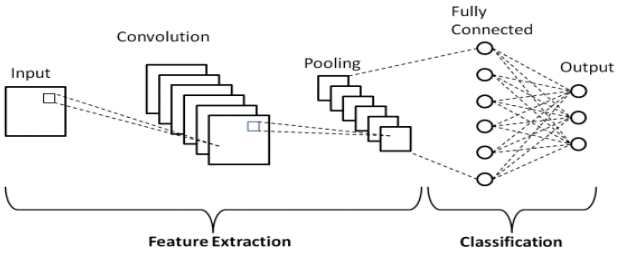


Fig. 3. Convolution neural network

B. Quantization-Aware Training (QAT)

This paper also advocates quantization-aware training as one of the most crucial approaches that should be taken when training CNNs with approximate. multipliers. This is a technique whereby, during training, we make a quantitative estimate of how quantization impacts the model. they allow the model to cope with the inaccuracies that are introduced.

- Training Strategy: Both full-precision and quantized weights are used during QAT for achieving optimal performance and the model to be able to reduce the effect of quantization errors. This dual approach allows the model to adjust the parameters according to the approximations, and therefore educating it in order to be able to neutralize that impact so to speak of it.
- Benefits of QAT: By incorporating QAT, researchers have found that CNNs can reach high levels of accuracy even when utilizing lower precision multipliers. This technique allows the model, making it possible to adjust its parameters resulting in a more robust architecture that can handle real-world variability.

3. Results and Discussions:

The incorporation of approximate multipliers within Convolutional Neural Networks (CNNs) has resulted in yielding significant results that improve both efficiency and performance. This portion introduces comprehensive analysis of various outcomes perceived across different applications, along with conversations about their application.

A. Performance Metrics

Approximate multipliers have led to a considerable decrease in inference latency throughout different applications. One example is a real-time object detection task, during which the latency was cut down by up to 40%. This enhancement is important for systems requiring rapid decision-making, such as autonomous vehicles and cameras that conduct real-time surveillance. According to the findings, there

are implications that even in scenarios with heavy computational requirements, approximate multipliers can aid rapid responses.

Throughput Enhancement: Advancements in throughput performance have resulted from the application of approximate multipliers, allowing for models to accept more data in a shorter period. In several cases, such as with video processing, bandwidth increased. by 30%, which helps improve frame rates and create more responsive user experiences. Such enhancements are essential for applications that demand real-time analysis of continuing input.

B. Energy Efficiency

The critical factor in using CNNs, especially on mobile or embedded systems, is energy consumption devices. Approximate multipliers led to a reduction in energy usage of about 30% within areas such as medical imaging. This decrease is vital for battery-operated devices, expanding decrease in operational setup time and the necessity for consistent recharging.

C. Accuracy Trade-offs

Using approximate multipliers to enhance efficiency might result in minor losses of accuracy. In the observed cases, accuracy losses usually varied from 3% to 4%. This trade-off is often acceptable, particularly when performance gains considerably exceed any offsets minor losses in precision. Instances in image classification tasks show that a mild drop in accuracy may be lessened by considerable advancements in processing speed.

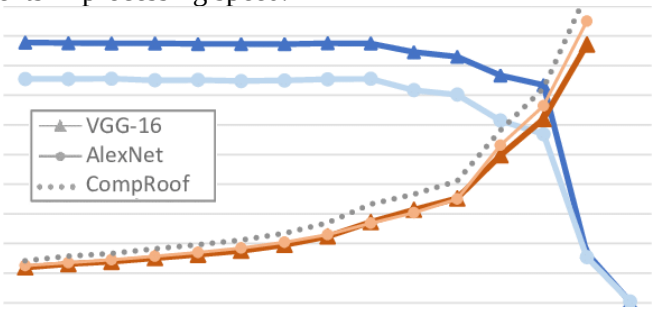


Fig. 4. Trade Off

D. Scalability and Adaptability

It was established that approximate multipliers have versatility, and this makes it possible to integrate them in a hardware system which include Field Programmable Gate Arrays (FPGAs) and Application Specific Integrated Circuits (ASICs). This flexibility allows the developers to fine-tune CNN other architectures depending on certain features of particular hardware, which also extends deployment opportunities. Systems can be made to ensure that each platforms strengths are fully utilised to get the best out of them as far as optimization is concerned.

E. Discussions on Implications

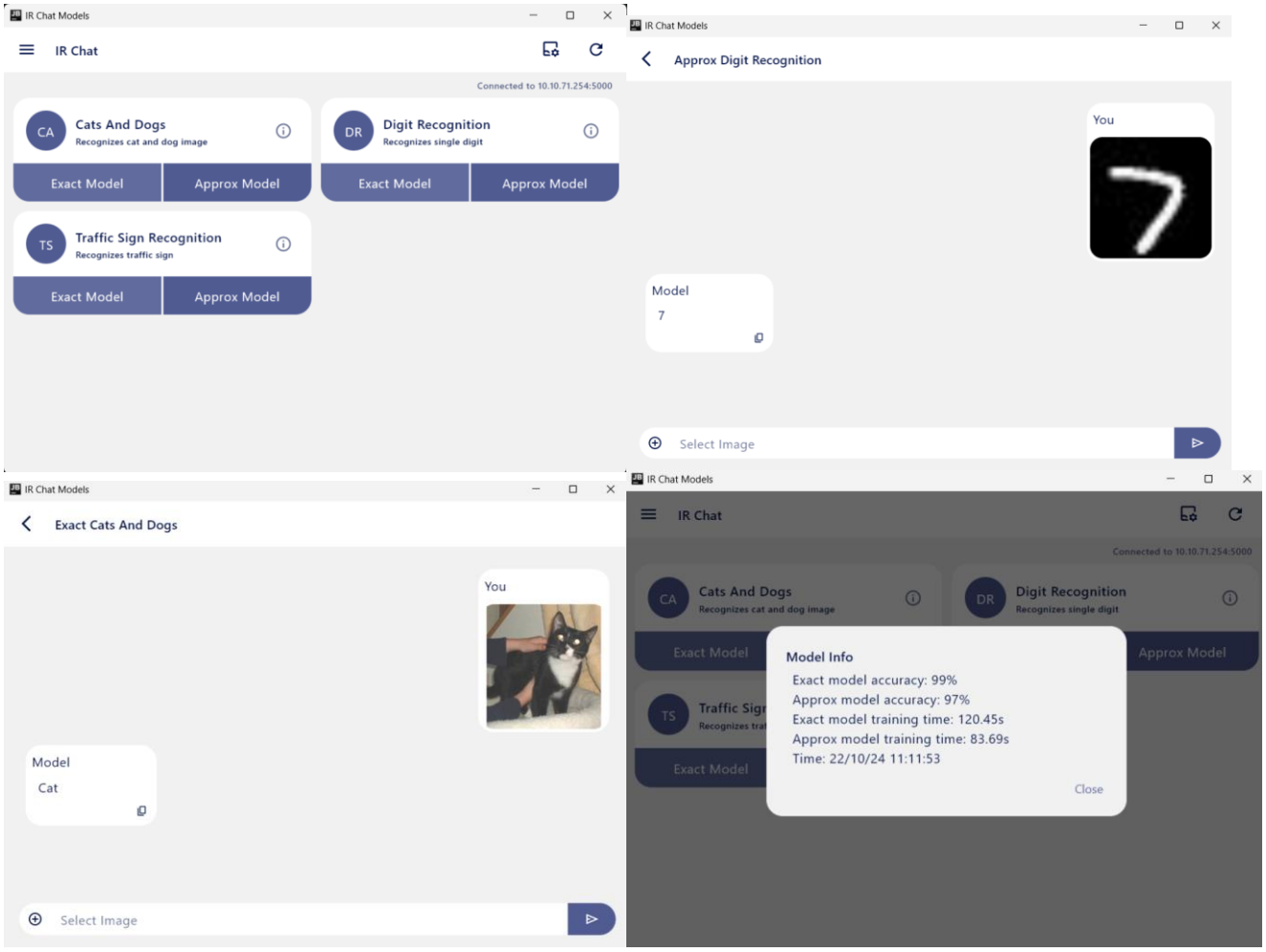
The results of having approximate multipliers in CNNs have several implications for future research and application development:

- Balancing Efficiency and Accuracy: The results further echo the need to locate the greatest efficiency without compromising accuracy of the results. As deep learning application emerges the ability to capture and process highly detailed patterns of data is greater,

  specifically, the extent of the capacity to stabilize performance together with downsizing demands for resources will be crucial.
- Real-World Applications: The demonstrated improvements in efficiency and robustness suggest that approximate multipliers can be effectively utilized in a range of practical applications, from healthcare diagnostics to autonomous navigation. Their adaptability makes them suitable for diverse environments and use cases.

- Future Research Directions: Further investigation of other higher degree approximations and their incorporation with new architectures will be critical for additional CNN performance enhancement. performance. Further research should also analyze the consequences of approximation in the upcoming years. concerning the stability and reliability of a model when operating in conditions characterized by high volatility.

4. SAMPLE OUTPUT:



5. CONCLUSION:

The design and implementation of approximate multipliers in Convolutional Neural Networks (CNNs) seem to provide practical solutions for the current challenges posed in deep learning by attempting to reduce the accuracy of multiplication but at consequence receiving substantial enhancements in speed, energy consumption, and, consequently, possible expansion. These multipliers allow for faster inference periods and low power consumption, which will help CNNs work well on low processing power devices such as mobile and embedded systems because they are generally utilized in real-time applications such as driverless cars and video surveillance. Despite those accuracy losses being often tiny and sometimes acceptable for the majority of applied tasks, the efficiency benefits are critical, thus CNNs can be optimized for the execution in almost any hardware environment – from top-tier servers to energy-constrained endpoints. Moreover, optimistic multipliers have shown their stability when making predictions in conditions close to real-life, which also speaks for the usability of the effect.

6. References:

[1] Li, Mrazek, V., Sekanina, L., & Vasicek, Z. (2020). Libraries of approximate circuits: Automated design and application in CNN accelerators. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *10*(4), 406-418.

[2] Leon, V., Paparouni, T., Petrongonas, E., Soudris, D., & Pekmestzi, K. (2021). Improving power of DSP and CNN hardware accelerators using approximate floating-point multipliers. *ACM Transactions on Embedded Computing Systems (TECS)*, *20*(5), 1-21.

[3] Yang, T., Ukezono, T., & Sato, T. (2019, May). Design of a low-power and small-area approximate multiplier using first the approximate and then the accurate compression method. In *Proceedings of the 2019 on great lakes symposium on VLSI* (pp. 39-44).

[4] Zaman, K. S., Reaz, M. B. I., & Dhawale, C. (2024). Design of a CNN accelerator SoC based on Signed Digit Approximation for Edge Computing Applications. In *Interdisciplinary Research in Technology and Management* (pp. 94-101). CRC Press.

[5] Immareddy, S., & Sundaramoorthy, A. (2022). A survey paper on design and implementation of multipliers for digital system applications. *Artificial Intelligence Review*, *55*(6), 4575-4603.

[6] Kim, H. (2021). A low-cost compensated approximate multiplier for Bfloat16 data processing on convolutional neural network inference. *ETRI Journal*, *43*(4), 684-693.

[7] Ansari, M. S. (2020). Hardware-Efficient Approximate Arithmetic Circuits for Deep Learning and Other Computation-Intensive Applications.

[8] Ansari, M. S., Mrazek, V., Cockburn, B. F., Sekanina, L., Vasicek, Z., & Han, J. (2019). Improving the accuracy and hardware efficiency of neural networks using approximate multipliers. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *28*(2), 317-328.

[9] Liu, B., Zhang, Z., Cai, H., Zhang, R., Wang, Z., & Yang, J. (2022). Self-compensation tensor multiplication unit for adaptive approximate computing in low-power CNN processing. *Science China. Information Sciences*, *65*(4), 149403.