Accelerating Artificial Neural Networks on Embedded FPGA with Hybrid Custom Floating-Point and Logarithmic Dot-Product Approximation

Yarib Nevarez
Ph.D. candidate at Universität Bremen

ABSTRACT

We accelerate artificial neural networks (ANN) optimizing the floating-point computation with vector dot-product approximation. The proposed method exploits the intrinsic error resilience of machine learning (ML) algorithms to reduce computational latency, memory footprint, and power dissipation while preserving inference accuracy. To demonstrate our approach, we address a hardware design exploration with convolutional neural networks (CNNs) and spiking neural networks (SNNs).

Introduction

We accelerate ANN with a dot-product hardware design based on approximate computing with hybrid custom floating-point and logarithmic number representation. This hardware unit has a quality configurable scheme based on the bit truncation of the synaptic-weight vector. Fig. 1 illustrates the dot-product hardware module with standard floating-point (IEEE 754) arithmetic, and our approach with hybrid custom floating-point as well as logarithmic approximation. As a design parameter, the mantissa bit-width of the weight vector provides a tunable knob to trade-off between efficiency and quality of result (QoR)[1]. Since the lower-order bits have smaller significance than the higher-order bits, truncating them may have only a minor impact on QoR [2]. Further on, we can remove completely the mantissa bits in order to use only the exponent of a floating-point representation. Therefore, the most efficient setup and yet the worst-case quality configuration becomes a logarithmic representation, which consequently leads to significant architectural-level optimizations using only adders and shifters for dot-product approximation in hardware. Moreover, since approximations and noise have qualitatively the same effect[3], we apply noise tolerance plots as an intuitive visual measure to provide insights into the quality degradation of ANN under approximate processing effects.

Dot-Product Hardware Block

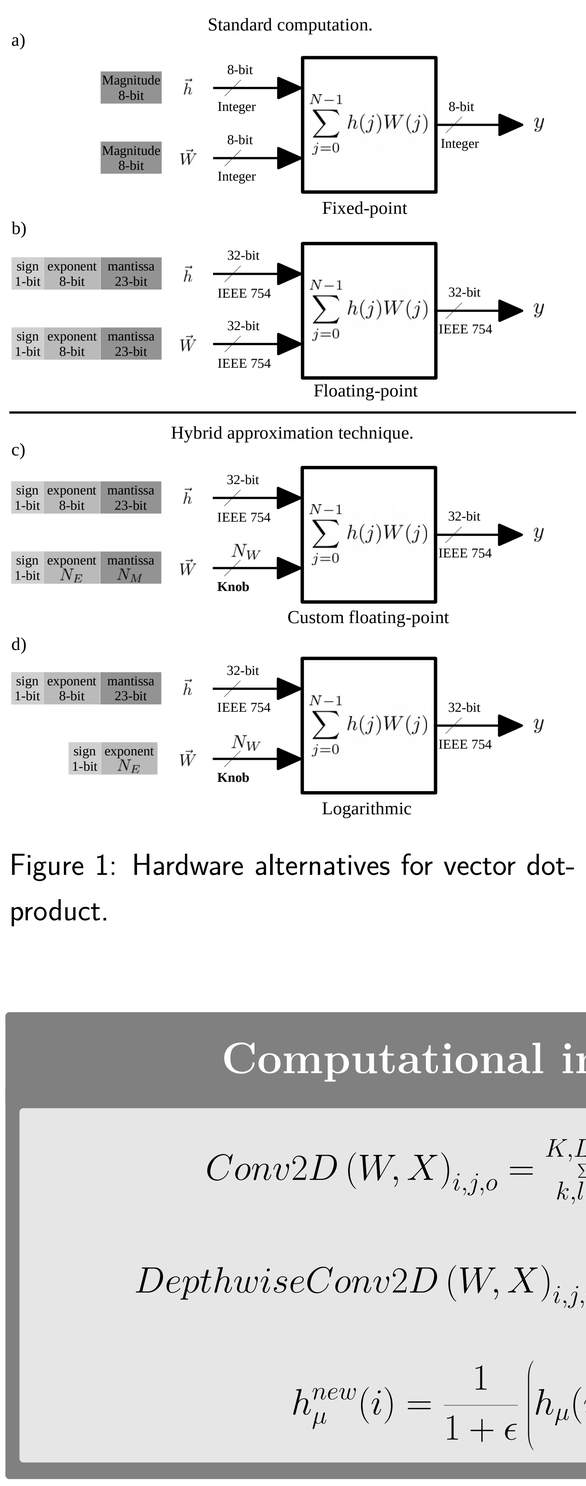

Figure 1: Hardware alternatives for vector dot-product.

Computational intensive operations

$$Conv2D\,(W,X)_{i,j,o} = \sum_{k,l,m}^{K,L,M} W_{(o,k,l,m)} \cdot X_{(i+k,j+l,m)+b_o} \quad (1)$$

$$DepthwiseConv2D\,(W,X)_{i,j,n} = \sum_{k,l}^{K,L} W_{(k,l,n)} \cdot X_{(i+k,j+l,n)} + b_n \quad (2)$$

$$h_\mu^{new}(i) = \frac{1}{1+\epsilon}\left(h_\mu(i) + \epsilon \frac{h_\mu(i)W(s_t|i)}{\sum_j h_\mu(j)W(s_t|j)}\right) \quad (3)$$

Embedded System Platform

In this section, we present a tensor processor compatible with TensorFlow Lite to accelerate *Conv2D* and *DepthwiseConv2D* operations on embedded FPGA. This implementation is integrated in a hardware/software co-design framework to accelerate tensor operations on FPGAs.

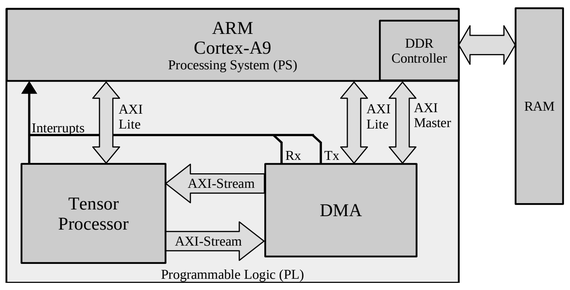

Figure 3: Embedded system architecture of the proposed compute platform.

Tensor Processor

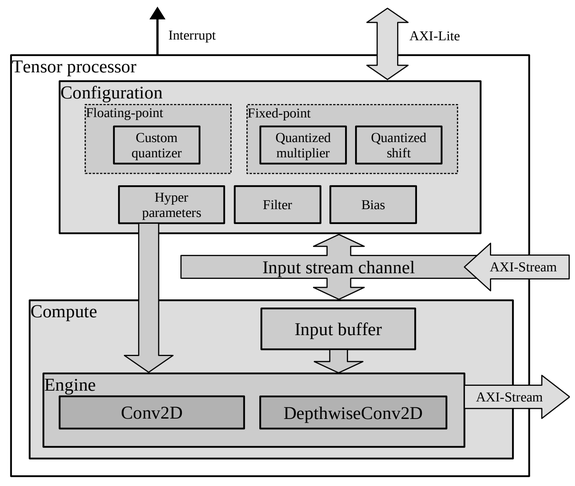

Figure 2: Embedded system architecture of the proposed compute platform.

Results

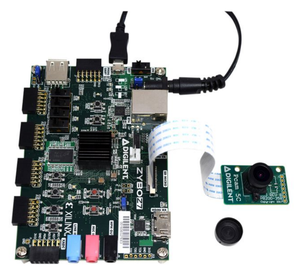

Figure 4: Figure caption

Nunc tempus venenatis facilisis. Curabitur suscipit consequat eros non porttitor. Sed a massa dolor, id ornare enim:
Nunc tempus venenatis facilisis. Curabitur suscipit consequat eros non porttitor. Sed a massa dolor, id ornare enim:
Nunc tempus venenatis facilisis. Curabitur suscipit consequat eros non porttitor. Sed a massa dolor, id ornare enim:
Nunc tempus venenatis facilisis. Curabitur suscipit consequat eros non porttitor. Sed a massa dolor, id ornare enim:

Spike-by-Spike neural networks

SbS is a spiking neural network approach based on a generative probabilistic model. It iteratively finds an estimate of its input probability distribution $p(s)$ (i.e. the probability of input node s to stochastically send a spike) by its latent variables via $r(s) = \sum_i h(i)W(s|i)$. where $\vec{h}$ is an inference population composed of a group of neurons that compete with each other.

Additional Information

Maecenas ultricies feugiat velit non mattis. Fusce tempus arcu id ligula varius dictum.

- Curabitur pellentesque dignissim
- Eu facilisis est tempus quis
- Duis porta consequat lorem

Maecenas ultricies feugiat velit non mattis. Fusce tempus arcu id ligula varius dictum.

- Curabitur pellentesque dignissim
- Eu facilisis est tempus quis
- Duis porta consequat lorem

References

[1] Jie Han and Michael Orshansky. Approximate computing: An emerging paradigm for energy-efficient design. In *2013 18th IEEE European Test Symposium (ETS)*, pages 1–6. IEEE, 2013.

[2] Sparsh Mittal. A survey of techniques for approximate computing. *ACM Computing Surveys (CSUR)*, 48(4):1–33, 2016.

[3] Swagath Venkataramani, Srimat T Chakradhar, Kaushik Roy, and Anand Raghunathan. Approximate computing and the quest for computing efficiency. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2015.

Acknowledgements

Nam mollis tristique neque eu luctus. Suspendisse rutrum congue nisi sed convallis. Aenean id neque dolor. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Contact Information

- nevarez@item.uni-bremen.de
- linkedin.com/in/yarib-nevarez