SOLVE THE IMPOSSIBLE

Low-Power Deep Neural Networks for IoT Edge Sensors

Abstract

This project will investigate novel DNN techniques for time-series data and its hardware implementation. The project will address the challenges of implementing complex DNN models for energy efficient and performance driven IoT edge devices. The project aims to develop hardware architectures that are scalable, and easily adaptable to different DNN models and applications. The DNN architectures developed will be demonstrated in hardware for a wearable healthcare application such as heartbeat identification.

Research Challenges

In traditional systems, the data generated in IoT devices are sent to cloud servers over wireless networks. AI and deep learning techniques are employed on servers for data processing and analytics. Due to the large amount of data generated in these sensors, it is beneficial (for system power, bandwidth) to process this data locally at the IoT node and send the inference to the cloud instead. However, there are several challenges here such as 1) many existing DNNs are not optimised for time series data from IoT devices 2) training phase of DNNs demands high computational power 3) hardware implementation of DNNs require significant resources.

In this project, we plan to

1) develop DNNs specifically targeted for time-series data using 1D convolution
2) do the training process offline and only do the inferencing on IoT device

3) develop circuit architectures that are scalable and programmable for low power and low complexity
4) verify the entire system on an FPGA.

For developing the DNN model, we plan to use a connected health use case of heartbeat identification from biomedical data. Although the DNN model developed will be specific to an application, the hardware will be implemented in such a way that it is scalable and programmable to support other applications as well.

For reducing the complexity of DNN hardware, we plan to use a combination of algorithmic and circuit approaches such as

1) optimizing the DNN topology for reducing model size
2) applying network pruning technique to remove insignificant connections in the DNN model
3) binarization/ quantization of the weights and/or intermediate signals in the DNN model
4) use low precision approximate circuit to reduce hardware cost at the expense of accuracy
5) optimizing memory access schemes to reduce the loss of performance associated with memory access

The above techniques are proven in literature as standalone approaches for specific applications. In this project all the above techniques will be implemented at once while making the DNN hardware programmable. For different applications, the DNN models and its parameters such as depth, size etc will be different. Therefore, design-time parameterisation will be used for making the DNN hardware easily scalable and flexible with respect to size, parameters and selection of optimization techniques and compile to the minimum possible hardware. This will enable the hardware to be adaptable to different applications. The above techniques may potentially reduce the accuracy of the DNN model while reducing the complexity. Therefore, a careful analysis on the trade-off of complexity vs accuracy will be performed. The DNN hardware will be implemented on an FPGA and will support convolutional neural network (CNN) type architectures. Extensive experimentation and evaluation will be performed to validate and tune the proposed architecture for a specific application.

Update

We have made job offers to 1 post doc candidate, who is expected to join UCD in 2 months' time. One PhD student funded by SFI ML-Labs is currently working on this project. The PhD student is working on developing a DNN accelerator fabric which can be integrated with a RISC-V CPU and could be used for deploying CNN based models. We have developed a CNN based deep learning model for detection of Atrial Fibrillation from electrocardiogram signals. This model will be used for demonstrating DNN hardware in FPGA. The developed model demonstrates high performance despite being trained on limited, variable-length input data. Weight pruning and logarithmic quantization are combined to introduce sparsity and reduce model size, which can be exploited for reduced data movement and lower computational complexity.