Modulation Classification for RFSoC showcasing Streaming-CNN Architectures

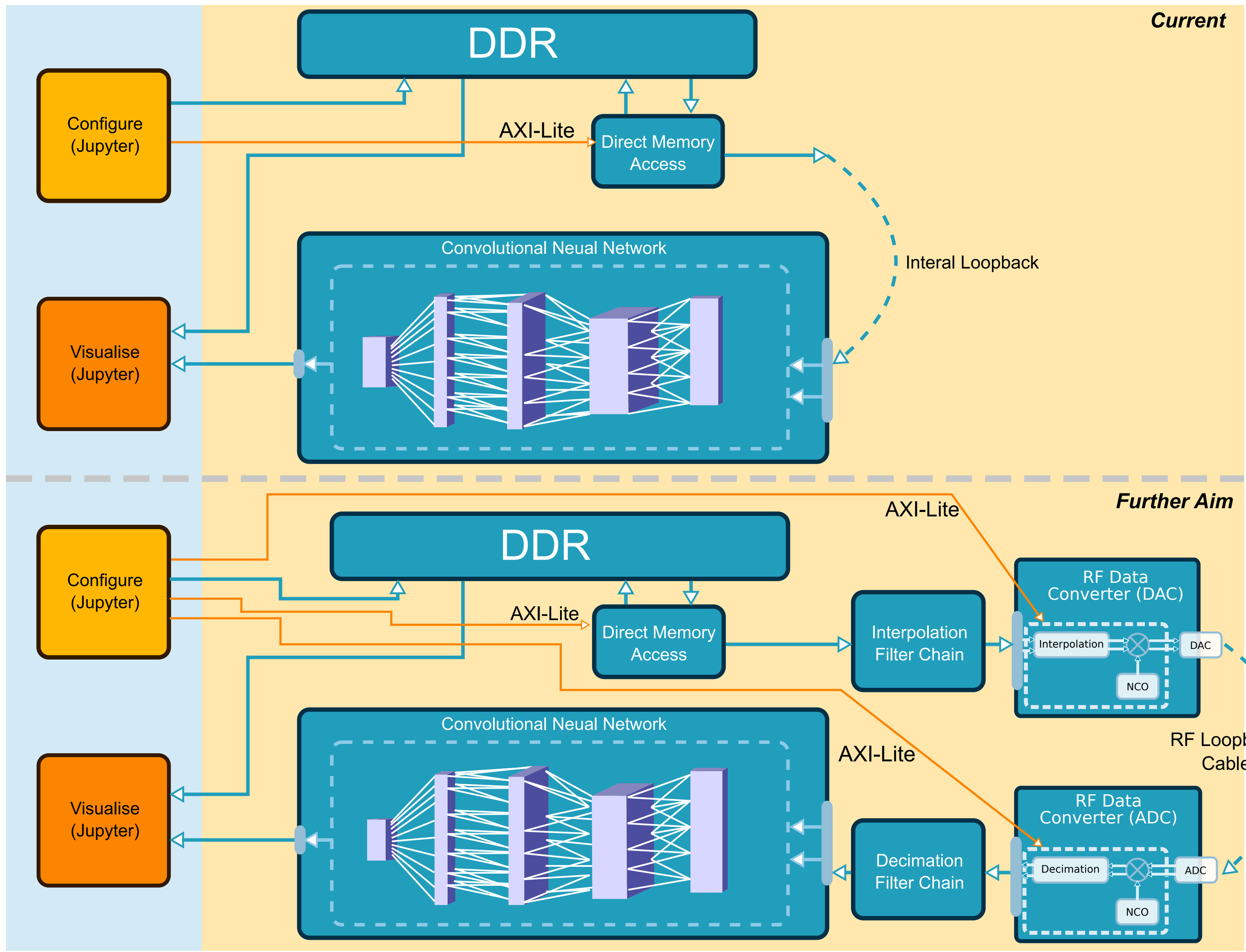
Andrew Maclellan, Louise H. Crockett, and Robert W. Stewart
Department of Electronic and Electrical Engineering, University of Strathclyde

Introduction

In wireless communications, the amount of available **spectrum** is reducing as the demand for connectivity increases. The increase of devices such as mobile phones, Internet of Things, smart vehicles, and wearable devices are making sections of the radio spectrum quite congested. **Shared spectrum** is an approach that aims to optimise the use of wireless communications channels to be shared among multiple users. A core aspect of shared spectrum is spectrum sensing, where a radio device detects other users within the nearby channels in order to transmit with minimal interference. Knowledge of how data has been transmitted from nearby users can assist in the transmission decisions for a radio device. In this work, we present a **modulation scheme classifier Convolutional Neural Network (CNN)** running on an AMD-Xilinx Zynq RFSoC 2x2 development board utilising a streaming-based architecture.

Our CNN is implemented entirely on the **Zynq RFSoC chip** including all weights and activations and uses 18-bit fixed point arithmetic. The user control and visualisation functionality is designed using the **PYNQ** software framework. The full system can be controlled on a web browser and interface with the development board using Python and interactive widgets. The project is open-source.

Application Overview



Our aim is to deploy an **18-bit quantised Convolutional Neural Network (CNN)** for **Automatic Modulation Classification (AMC)** that integrates with the PYNQ framework to control and visualise the input and outputs of the AI model.

AMC is a method of classifying the received samples on an antenna and identifying the modulation scheme the data is transmitted in.

AMC is a great challenge for showcasing out **streaming-CNN architecture** that is designed to process every sample received by the receiver with no interruptions.

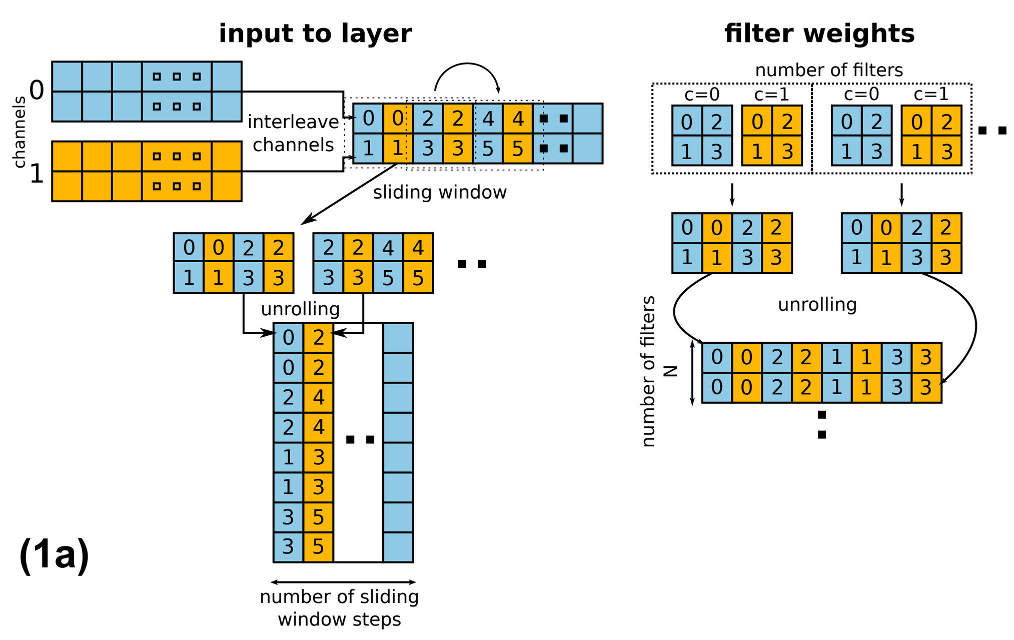
Currently, the streaming-CNN architecture has been tested on the internal logic fabric and we aim to further develop the system to include DAC transmission and ADC reception of modulation schemes.

CNN Architecture

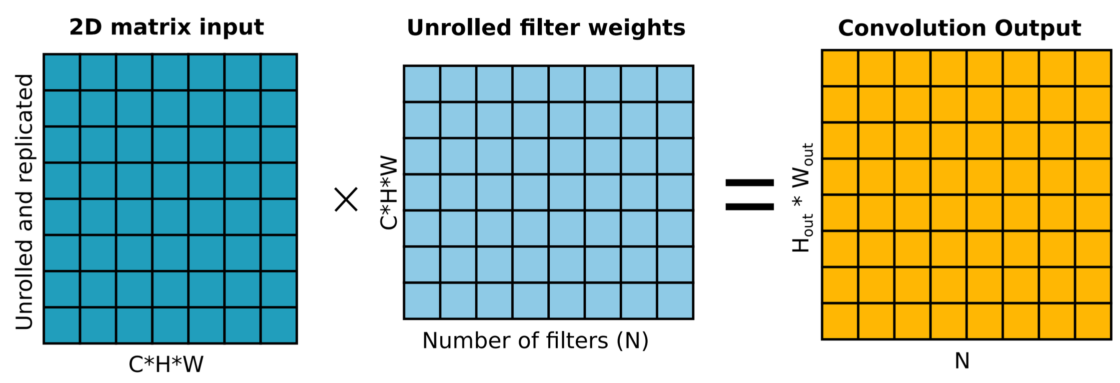
GEMM Transform (1a,1b): The **GE**neral **M**atrix-to-**M**atrix Transform is a technique used to reposition the incoming samples and filter weights to a simpler matrix-to-matrix multiplication. This is useful for FPGA deployment as this multiplication can be vectorised and parallelised.

Neural Network Layers (2): The AMC network is built up of four computational layers. Two convolutional layers used to find local correlations among samples and two fully-connected layers for more general decision making and analysis.

Streaming Architecture (3): Figure (3) overviews a high-level implementation of the AMC network on hardware utilising Matrix-to-Vector multiplications to calculate the convolutions and fully-connected layers. To convert the incoming samples into a GEMM matrix, the samples are stored in buffers and the resulting matrix is read out with the 'BRAM Controller'.



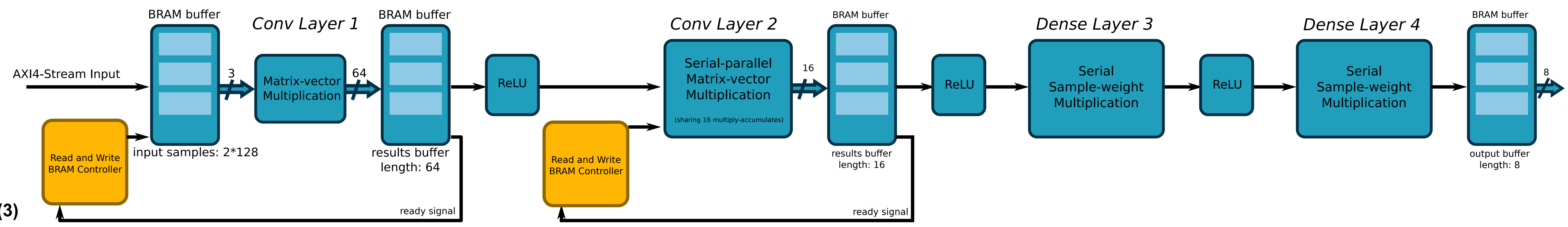
(1a)



(1b)

Layer #	Layer type	Neuron Dimensions	Activations	MACs
1	Input	2x128	-	-
2	Conv	64x1x3	ReLU	45354
3	Conv	16x2x3	ReLU	761556
4	Dense	128	ReLU	253952
5	Dense	8	Softmax	1024

(2)

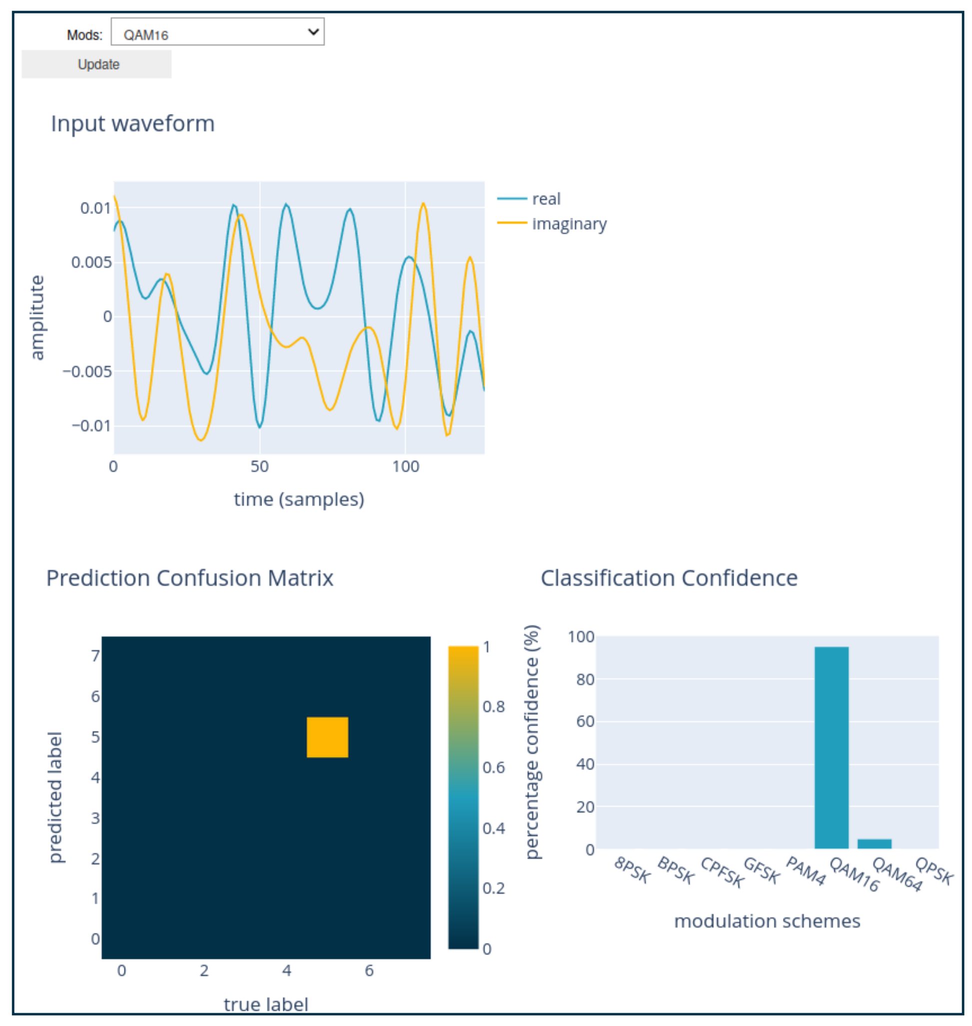


(3)

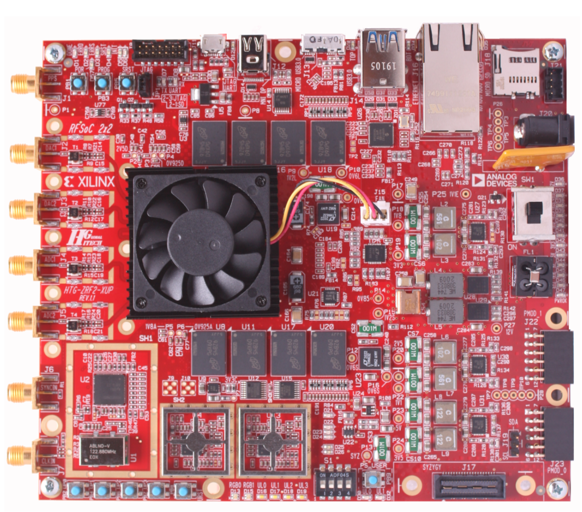
Interactive Modulation Selection

The AMC CNN can be tested through a Jupyter Notebook interactive displayed on a web browser with interactive controls to test modulation schemes and observe waveforms. The following figure was captured from a Jupyter session running on an AMD RFSoC 2x2 board. The neural network has been trained to identify the following modulation schemes (each modulation scheme has been pass-through a WINNER II channel model):

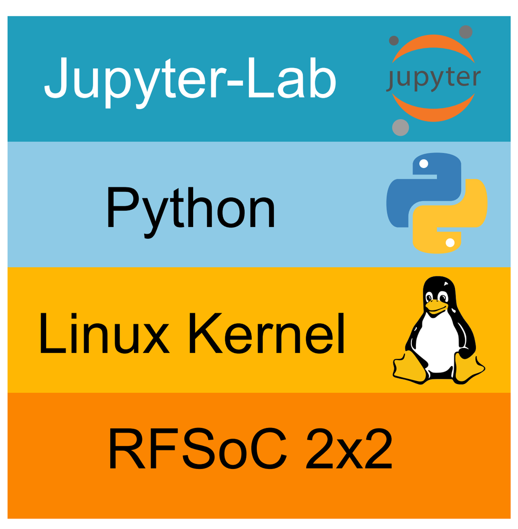
8PSK, **BPSK**, **CPFSK**, **GFSK**, **PAM4**, **QAM16**, **QAM64**, and **QPSK**.



PYNQ Framework & RFSoC



RFSoC 2x2



more info: pynq.io

Contact

github.com/axdy
in/andrew-maclellan
a.maclellan@strath.ac.uk

Performance

Average accuracy across all modulations is 94.1125%