Wireless Modulation Classification Using Deep Learning on FPGAs

Brenda So

October 16, 2017

Abstract

In our senior project, we aim to implement an FPGA module that can capture raw IQ samples over the air and perform modulation classification with convolutional neural networks. We will test the module in a controlled environment where IQ samples of different modulation schemes are sent out and intercepted, thus testing the robustness and effectiveness of CNNs in performing such a task.

This report presents research done on several aspects of our project. First, there is a brief discussion on convolution neural networks and an investigation of the dataset we wish to use to train the network. Second, research is done on Field Programmable Gate Arrays, specifically the Zynq7000, and its basic building blocks.

The last portion of this report focuses on the details of wireless communication and modulation classification. I first give a background on signal modulation and channel effects on transmitted signal. Afterwards, I present research done on state of the art techniques involved in modulation classification. Lastly, I present research on transmitter and receiver design and the possible use of software defined radio in our project.

1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a class of machine learning algorithms that models after the development of neurons. Contrary to traditional machine learning, where the user extracts specific features from the raw data to train the algorithm, CNNs learns the features as "neurons" from the raw data itself. What the user specifies in the CNN is the layers that the training data goes through at each stage. Example layers include fully connected layers, convolutional layers etc. The "convolutional" part of CNN comes from the common usage of convolution operations in its layers.

1.1 radioML

In order to train a CNN, we need large amounts of data. Such data is publicly available on radioML – an organization that provides simulated radio transmission data. [1] The data provided encompasses signals from 8 digital modulation schemes and 3 analog modulation schemes, including BPSK, AM-DSB, 8PSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, AM-SSB and WBFM. The simulated signals take into account realistic effects such as multipath fading, sample rate and center frequency offset. The SNR of these signals ranges from -20 dB to 18 dB.[3]

Although radioML is a useful dataset to train a CNN, it might not be sufficient to use it to test the CNN. It is because the dataset is synthetic, generated by the GNU radio dynamic channel modeling API.[2] This API incorporated effects of sample rate offset, center frequency offset, selective fading and noise into the simulated channel.[3] However, these effects are still simulated, so they could only be approximation of a physical channel at best. For instance, the simulated data did not take into account interference from other channels. Hence, the best way to test the usefulness of our CNN is to bring it to a physical environment.

2 Field Programmable Gate Arrays

Field Programmable Gate Arrays (FPGAs) are often used in digital design prototyping due to their customizability and relatively fast design flow. To program an FPGA, we use VHDL, a hardware description language used in digital design, to layout the circuit that the FPGA needs to emulate hardware. The software that we use to compile VHDL to bitstream is Vivado. The FPGA used in this project is the Zynq7000 series. The chip is mounted onto a board (a.k.a. Zedboard) with a variety of peripherals that allow us to prototype for different applications. The Zynq7000 chip has two major units – the processing system and the programmable logic.

2.1 Processing System (PS)

The processing system on the FPGA is a new addition to the traditional FPGA. It is an ARM Cortex-A9 processing unit. Such a unit has IP cores that allows us to interface with existing peripherals on the zedboard easily. e.g. There is a built-in DRAM controller on the processing unit to interface with on board DRAM. The user can choose to use or not use the PS.

2.2 Programmable Logic (PL)

The programmable logic on the FPGA is what is traditionally considered as an FPGA. There are 3 major components of the PL[4]:

- Look Up Tables (LUTs) : The look up table is an implementation of a truth table. On our FPGA, it takes in 6 input bits and outputs 1 bit.
- Flipflops (FFs) : Each LUT can be optionally connected to an FPGA. 4 LUTs and 8 FFs form a slice
- Routing Matrix : Two slices are connected to a configurable logic block (CLB). The CLB is connected to a switch matrix (composed of a series of multiplexers)that connects the input and output of CLBs to each other, forming a complete circuit.

With the LUTs, FFs and Routing matrix, Vivado runs a series of proprietary synthesis and implementation algorithms that translates VHDL to a digital circuit. However, the resources available on the FPGA is limited. Hence, one of the most challenging problems of the project is to fit the CNN, which usually consists of a lot of layers and features, onto one FPGA.

3 Signal Modulation

When transmitting signals, it is beneficial to modulate the original signal with a high frequency periodic carrier signal. The periodic property allows the receiver to demodulate and recover information easily. Current modulation schemes can be classified into two classes:

- Analog Modulation - Transfer an analog baseband signal over an analog channel at a higher frequency. Examples include AM (Amplitude Modulation) and FM (Frequency Modulation) radio.

- Digital Modulation - Transfer a bitstream signal over an analog channel by encoding the bitstream with an analog carrier. Examples include Binary Phase Shift Keying (BPSK) modulation and QAM (Quadrature Amplitude Modulation) schemes.

A generic received signal can be represented by the following equation :

$$r(t) = c(t) * s(t) + n(t) \tag{1}$$

where r(t) is the received signal, s(t) is the transmitted signal, n(t) is the additive white gaussian noise (AWGN) and c(t) is the time-varying impulse response of the transmitted wireless channel. The goal of our classifier is to predict the modulation class that s(t) belongs to with r(t) as the only reference. The transmitted and received signal is commonly represented in IQ form, where I represents the real part of the signal and Q represent the imaginary part of the signal. In our implementation, the signal will first be modulated by a fixed modulation scheme and then converted to raw IQ data with a DAC. At reception, the raw IQ samples will be converted back to normal representation with an ADC.

3.1 Channel Effects and Recovery

A transmitted signal often experience a number of effects from the physical environment[5]. Some examples include:

- AWGN - Noise in the environment (e.g. other EM waves) makes the transmitted signal noisy
- Multipath Fading - As signal travels, the signal might reflect off different objects and arrive at the receiver. These signal interfere with each other.
- Carrier Frequency Offset - If the receiver and the transmitter is moving, the doppler effect can lead to a mismatch in frequency.

Out of the aforementioned channel effects, the only linear effect is AWGN. Multipath fading and carrier frequency offsets are non-linear.

On the receiving end of a radio communication system, a matched filter is commonly used to recover the symbol. Expert designed filters are used to convolve over the received signal, forming peaks when the matching symbol is found.[6] However, by using a match filter, we assume that users know what signal and what modulation scheme they are going to receive (hence knowing which expert designed filter to use). Moreover, a matched filter is only optimal in the presence of Guassian Noise. Other effects, particularly non-linear effects, are not taken into account by a matched filter.

One possible use of a CNN is to take into account non-linearity of the received signal. There are network layers that are built to add non-linearity to the overall CNN architecture. A commonly used non-linear unit is the rectified linear unit (ReLU) [7], which gives an output of 0 if the input is negative and the same value otherwise.

4 State of the Art

Before the rise of CNN, previous research in modulation classification can be generalized into two classes : likelihood based and feature based.

Likelihood based methods are based on a pre-defined likelihood function of the received signal. A decision on the type of modulation scheme is reached by performing a likelihood ratio test. [8] Although this method is optimized from a Bayesian point of view, i.e. it can minimize the probability of false classification, it has a very high computational complexity, hence increases the latency of classification results. [8]

Feature based methods uses expert features and reaches decisions based on observed values. Such features could include normalized signal amplitudes, phase, frequencies, variance of zero-crossing intervals etc.[8] With said features, we could run traditional machine learning algorithms, such as Support Vector Machines (SVM), to detect the modulation schemes. However, expert features that requires data collection over long periods of time are hard to obtain due to the short time nature of the collected signals. [5]

5 Radio Design

In order to test the FPGA module in a physical environment, we need to setup radios that can send and receive signals. This leads to research into radio design and the use of software defined radios (SDR) to setup such an environment.

5.1 ISM Band

The Industrial, Scientific and Medical (ISM) band are bands that allow unlicensed users to transmit wireless signals through. The standard that we are adhering to is the 802.11g standard [9], encompassing the following requirements:

- Maximum Data Rate : 54 Mbps
- Modulation : OFDM, CCK, DSSS
- RF Band : 2.4 GHz
- Channel Width : 20 MHz

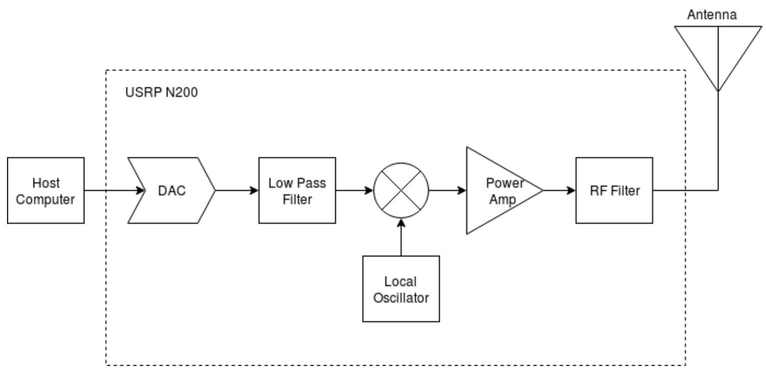

Figure 1: Radio Transmitter Block Diagram [10]

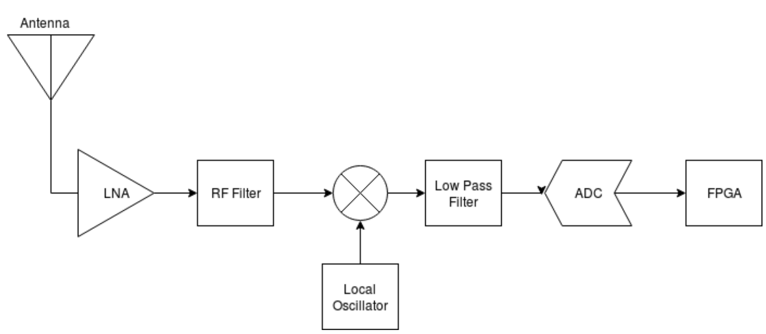

Figure 2: Radio Receiver Block Diagram [10]

5.2 Traditional Radio Design

Figure 1 shows a top-level overview of a transmitter. The modulation scheme is chosen and implemented by the host computer. After converting the digital signal to analog with the DAC, the signal is passed through the low pass filter as an anti-imaging filter. The mixer and the local oscillator brings the baseband signal to the 2.4 GHz band. The power amplifier amplifies the signal and the final signal is broadcasted by the antenna. This overall layout is useful in implementing a controllable transmitter, where we can select different modulation schemes and transmit a fixed signal.

Figure 2 shows the top-level overview of a receiver. After receiving the signal from the antenna, the low noise amplifier (LNA) will amplify the signal without perturbing the signal too much. The RF filter filters out noise at higher frequencies and the oscillator brings the high frequency signal down to baseband. The low pass filter in the receiver is an anti-aliasing filter and the ADC converts the analog signal to digital form, which is then processed by the FPGA for modulation classification.

5.3 Software Defined Radio

A software defined radio (SDR) is a radio with reconfigurable hardware, commonly implemented with an FPGA. While an analog radio requires hardwiring components to one another, SDR offers great flexibility in building a radio. While users can still alter the frequency and bandwidth of the transmitted signal, it abstracts the hardware away and allows users to focus on experimenting with the radio instead of building the radio.

The USRP N200 is a readily available SDR. In our environment, the SDR is controlled by a host computer to transmit 11 different kinds of modulation scheme (the ones available in the radioML dataset). Its operating frequency ranges from 0 MHz to 6 GHz, which overlaps the 802.11 2.4 GHz band. Its bandwidth is 25 MHz for 16-bit samples and 50 MHz for 8-bit samples. [11] The SDR can be programmed with GNU Radio – an open source SDR development platform. For instance, the component required to satisfy the 802.11g requirements is shown in Figure 3. The figure shows that This implementation is much simpler than physically building a transmitter and receiver that can interface with a host

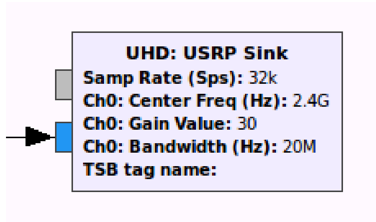

Figure 3: Interface from GNU Radio to USRP N200

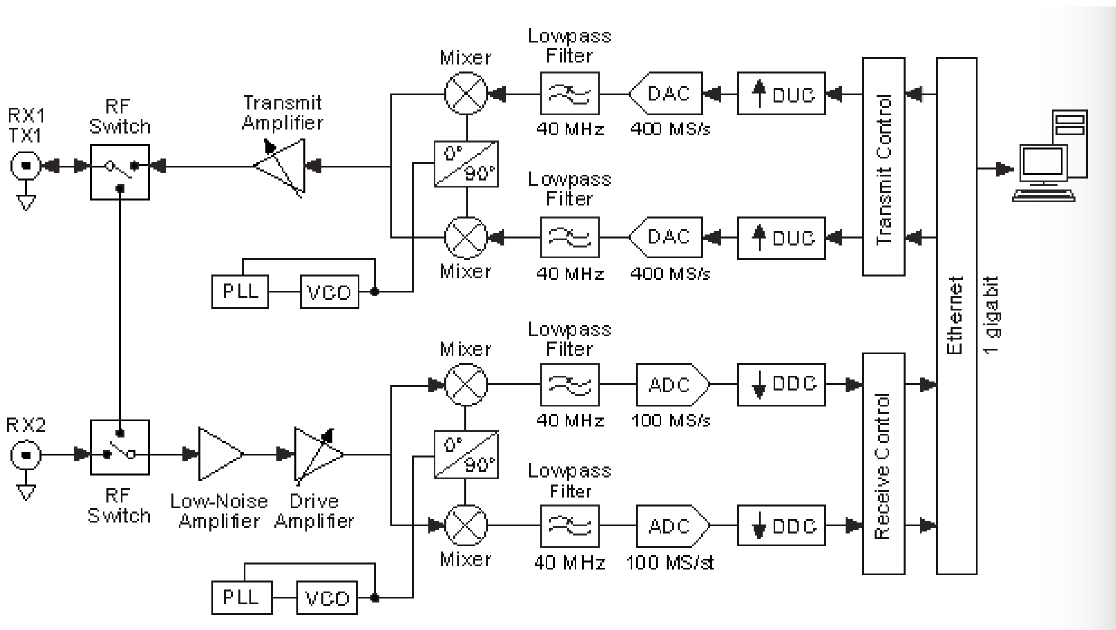

Figure 4: General Architecture of radio inside USRP N200[12]

computer and the FPGA, thus allowing us to spend more time in developing the CNN on the FPGA.

The general architecture of the radio inside USRP N200 is shown in Figure 4. All components are components built by Ettus Research, and its structure mirrors that of an analog radio. The two paths for each control reflects the presence of I and Q samples in the radio. The sample rate of the ADC and DAC on the board determines the FPGA processing bandwidth (for N200 it is 100 MS/s for ADC and 400 MS/s for DAC) inside the SDR. [12]

References

[1] "RadioML – Advancing Radio Systems that Learn from Data and Approximate the Complexity of the Real World," 2016, https://radioml.com/.

[2] N. W. Timothy J O'Shea, "Radio Machine Learning Dataset Generation with GNU Radio," in *Proceedings of the GNU Radio Conference*, vol. 1, 2016.

[3] T. J. O'Shea, "radioML Dataset," 2016, https://github.com/radioML/dataset.

[4] "How does an FPGA work?" https://embeddedmicro.com/tutorials/mojo-fpga-beginners-guide/how-does-an-fpga-work.

[5] J. H. Timothy J O'Shea, "An Introduction to Deep Learning for the Physical Layer," *CoRR*, vol. abs/1702.00832, 2017. [Online]. Available: http://arxiv.org/abs/1702.00832

[6] T. J. O'Shea and J. Corgan, "Convolutional radio modulation recognition networks," *CoRR*, vol. abs/1602.04105, 2016. [Online]. Available: http://arxiv.org/abs/1602.04105

[7] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," *CoRR*, vol. abs/1611.01491, 2016. [Online]. Available: http://arxiv.org/abs/1611.01491

[8] O. A. Dobre, A. Abdi, Y. B. Ness, and W. Su, "Survey of Automatic Modulation Classification Techniques: Classical Appraoches and New Trends," *IET Communications*, vol. 1, pp. 137–156, 2007.

[9] I. Poole, "IEEE 802.11g Wi-Fi Tutorial," www.radio-electronics.com/info/wireless/wi-fi/ieee-802-11g.php.

[10] V. A. Sergeev, "LuaRadio," luaradio.io/new-to-sdr.html.

[11] "USRP Hardware Driver and USRP Manual," files.ettus.com/manual/page_usrp2.html.

[12] "About USRP Bandwidths and Sampling Rates," https://kb.ettus.com/About_USRP_Bandwidths_and_Sampling_Rates.