Application of Deep Learning in Wireless Signal Identification for Intelligent Channel Sensing

Md Faisal Kabir1, Ahmad Y. Javaid1, and Vijay Devabhaktuni2
1Dept. of Electrical Engineering and Computer Science, The University of Toledo, OH, USA
2Dept. of Electrical and Computer Engineering, Purdue University Northwest, Hammond, IN, USA
Email: {mdfaisal.kabir, ahmad.javaid,}@utoledo.edu, vjdev@pnw.edu

Abstract—Evolution in technology demands connectivity as a basic requirement, but the frequency spectrum is not an abundant resource. Cognitive radio and cognitive network have been developed to address the under-utilization of the licensed frequency and make it available to use for the unlicensed user. NASA has also adopted cognitive technology for its space communication. The cognitive system relies upon its channel sensing system to make decisions. Our work focused on adding to the channel sensing capability which is a deep neural network approach to recognize the signal modulation type. Along with the NASA SCaN testbed dataset, we trained and tested the neural network model. The results have proven promising in the application for signal identification.

Keywords: intelligent channel sensing, deep learning, signal identification

1. Introduction

As technology is evolving at an unbelievable pace, today the need to stay connected is more than ever. Almost every device are utilizing the wireless network to fulfill the demand for connectivity. This utilization trend is creating a major effect on the radio spectrum by exhausting the resource. Unlicensed frequencies like Wifi is heavily used for the new trending internet of things which is adding to the wireless interference. However, many licensed frequency is not utilized to their full potential, which creates an opportunity to be used by the unlicensed user. FCC approved the idea for the unlicensed user to use TV-broadcast bands. This type of opportunistic utilization concept gave a whole new technology of Cognitive radio. Cognitive radio is such technology, which gives the unlicensed user access to the licensed frequency when the primary user (licensed user) is not present. Multiple numbers of CR connected create a Cognitive Network, where they share the resources following standard protocols. To use the frequency, the radios first need to sense the channel to determine the existence of the primary user. Channel sensing is a critical process for the Cognitive system not only for determining the purpose but also it can bolster the security of the CN.

To enhance the performance of the Space Communication and Navigation (SCaN) program, NASA has also employed the technology of CR and CN [1]. Zhang et al. have collaborated with NASA Glenn, to conduct a study with the cognitive radio and test certain concepts in the testbed. As cognitive radio primarily relies upon channel sensing, they have worked on secured cross-layer cognitive networking based on that.

In our work, we are proposing a modulation recognition based on the NASA SCaN testbed generated the dataset, which will add to the cognitive capability of the intelligent channel sensing system.

2. Related Work

Signal identification has been studied not only for cognitive network system also as fingerprinting in several security application. Boris et al. in their work [2] for wireless security for sensor nodes employed transient-based identification. They have acquired signal transient and transformed the samples using FFT spectra. In their work, they have used one dimensional Fourier transformation. Using Mahalanobis distance for the matching method, they trained their model which resulted in high accuracy for sensor node recognition. Selim et al. [3] used signal identification in spectrum monitoring for radar bands. They have used CNN(Convolutional Neural Network) which have been fed with IQ samples of the signal for training. They transformed the raw IQ samples into amplitude and phase. Utilizing deep learning for signal identification, more specifically signal modulation recognition is becoming well favored to the research community. Many works have shown promising results. Tim and Nathan in their work [4] used GNU Radio (Blossom,2014) to build a dataset for machine learning. This dataset was used then used for several other works like [5], where Timothy et al. have used CNN (convolutional neural network) to recognize the signal modulation. They used the complex time series input which has been transformed into a 2xN vector. The two columns have been created by the In-phase and quadrature value separately. They achieved 87.4% classification accuracy across all signal to noise ratios. In another work [6] they have implemented an OTAI (Over The Air) test-bed where they transmitted signals using USRP (Universal Software Radio Peripheral). Similarly, another USRP has been used as a receiver which produced training data for signal classification models. Three different, e.g. 1) Baseline Method, 2) CNN and 3) Residual Neural Network...
have been used for the classification problem. The baseline method has achieved 94.6% accuracy. Merima et al. [7] worked on end-to-end learning for modulation recognition and wireless interference identification. They have also used CNN, but they employed three types of data transformation and compared the results among three different training result. The comparison showed that the model worked better for wireless interference identification than modulation recognition. Tanzin et al. [8] have worked with NASA SCaN testbed data along with SDR generated data. They used ANN (Artificial Neural Network) to train the data. The algorithms used were Levenberg-Marquardt (LM) and Scaled Conjugate Gradient (SCG) algorithm.

In this paper, we have used Multi-Layer Perceptron (MLP) of the deep neural network which has been trained with NASA SCaN testbed dataset for modulation recognition. In the following sections, we have discussed the algorithms and methodologies we used to identify the signal modulation from the data.

3. Intelligent Channel Sensing

Zhang et al. [1] have proposed a simplified cognitive cycle which is appropriate for NASA SCaN system. In their opinion, this cycle addresses the unnecessarily complicated processes of the initial cognitive cycle which is mainly consisted of Observe, Orient, Plan, Decide, Act and Learn (OOPDAL Loop) [9].

![Image of simplified cognitive cycle](image)

Fig. 1: Simplified cognitive cycle reproduced from [1]

As shown in figure 1, the system first sense channel information. Based on the information it learns based on different machine learning methods. The cycle then advances toward adapting different protocol and waveform design that is best optimized for the channel. In their work, they also informed that NASA's most sophisticated radio use the Proximity-1 protocol, where Mars Science Laboratory (MLS) used Adaptive Data rate (ADR) for communication between Mars Relay Orbit (MRO) and Curiosity (MSL). This cognitive system is dependent upon sensing of the signal strength. Other significant parameters like modulation, center frequency, etc. if trained and stored by the channel sensing system, can enhance the decision making and the adaptability of the cognitive radios.

Understanding the requirements, we have worked on a deep neural network based learning system to successfully identify the signal modulation to enhance the intelligence of the sensing system further. The inspiration behind choosing deep neural network for the task is the nature of the problem. Modulation recognition is technically a classification problem. Deep neural network is known to be an efficient tool for such classification task. Training data have been collected from the NASA SCaN testbed and then processed by the signal processing techniques. In the next section, we discussed the types and methods of different parameters we used to build our deep neural network for this purpose.

4. Deep Neural Network

A neural network is a mapping equation which maps an input vector to an output vector consisted of different layers as 1) input layer, 2) hidden layer and 3) output layer. Every node in the layers is known as the neuron.

If there are L layers with an input vector \( r_0 \in \mathbb{R}^{N_0} \) which maps to output vector \( r_L \in \mathbb{R}^{N_L} \), then the mapping is a function described as \( f: \mathbb{R}^{N_0} \rightarrow \mathbb{R}^{N_L} \). The mapping is done by a process of iteration of L processing steps. The step can be defined as:

\[
    r_l = f_l(r_{l-1}; \theta_l), l = 1, \ldots, L
\]

Equation 1 is the \( l \)th layer mapping. Mapping is dependent on the output vector from the previous layer. Also, it is dependent on certain parameters which is denoted by \( \theta_l \). Some random variables are also part of the decision. This makes the function stochastic in nature.

4.1 Dense Layer:

The layers can be of different types. There are dense, noise, dropout and normalization layers. A layer is called a dense layer if it follows:

\[
    f_l(r_{l-1}; \theta_l) = \sigma(W_l r_{l-1} + b_l)
\]

in equation 2 \( \sigma \) is the activation function which is underlying in every neuron. the activation function addresses the non-linearity; without this, the concept of deep learning where one layer is stacked on another layer will not be a very effective one.

4.2 Activation function:

There are three types of activation function mostly used in deep learning systems. These are the sigmoid activation function 3, the Rectified linear unit (ReLU) function 12 and the leaky ReLU function. Leaky ReLU works similar to the ReLU function, but this function addresses the 'dying ReLU' problem. This problem occurs when a neuron is
stuck in negative side always resulting zero, also known as 'dead neuron.' Leaky ReLU function speeds up the training process, as it is more balanced than the ReLU function \[10\].

\[
\sigma(z) = \frac{1}{1 + e^{-z}} \quad (3)
\]

\[
\sigma(z) = max(0, z) \quad (4)
\]

We preferred the leaky ReLU function for the above said reasons in this training. The leaky ReLu function is also known as parameterized ReLU. The leaky ReLU function is as follow:

\[
\sigma(z) = 0.01z, \text{ when } z < 0 \quad (5)
\]

We have used the sigmoid activation function in our output layer. A sigmoid function is also known as the logistic activation function. The sigmoid function is described as:

\[
\sigma(z) = \frac{1}{1 + e^{-z}} \quad (6)
\]

The sigmoid activation function in equation 6 mainly addresses the problems of probability as the function is spread from 0 to 1. This nature of sigmoid function makes it a primary choice for an issue where we are getting probability as an output.

4.3 Adam Optimizer:

For our training purpose we have used the adam (adaptive moment estimation) optimizer algorithm. It is an efficient stochastic optimization technique because it has a low memory requirement. Adam optimizer uses the first and second moment of the gradient for estimation. it computes the previous decaying average and squared gradient \( m_t \) and \( v_t \) as following where \( \beta_1 \) and \( \beta_2 \) are the decay rates \[11\]:

\[
m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (7)
\]

\[
v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (8)
\]

Bias-corrected first moment and second raw moment are denoted as \( \hat{m}_t \) and \( \hat{v}_t \) and is computed as follow \[11\]:

\[
\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (9)
\]

\[
\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (10)
\]

Finally the adam parameter update rules as following \[11\]:

\[
\theta_t = \frac{\theta_{t-1} - \alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (11)
\]

4.4 Cost function:

We have used the binary cross entropy function as the loss calculation function for our training. The binary cross entropy function is as following:

\[
CE = -\sum y_i \log(f(z_i)) \quad (12)
\]

5. Dataset Description

NASA Scan testbed consists of several reconfigurable SDRs, one of which operates in Ka-band. We have the two sets of data of Ka-band of two different, i.e. BPSK and QPSK. The sampling frequency is 0.1MHz, and the symbol rate is 400Hz. The samples are in raw IQ format which is visualized in figure 2 and 3.

5.1 Method 1:

In this method \[5\], raw complex data \( C \) have been transformed to it’s two different component, the real part \( I \) and the imaginary part \( Q \). Hence the mapping is like below:

\[
C \rightarrow I, Q \quad (13)
\]

This operation changed the input vector dimension to 1600xN. We can see the visual representation of the different modulation data in figure 4 and 5.
5.2 Method 2:
Selim et al. [3] have shown that using amplitude and phase information of the modulated signal can provide a promising result. We have accommodated the idea to transform the raw IQ sample data into amplitude \(A\) and phase angle \(\phi\). From general complex signal mathematics we know the following equations 14 and 15 to calculate \(A\) and \(\phi\).

\[
A = \sqrt{I^2 + Q^2} \quad (14)
\]

\[
\phi = \tan^{-1} \frac{Q}{I} \quad (15)
\]

Visual representation of \(A\) and \(\phi\) is shown in figure 6 and 7. This representation method also resulted in input vector dimension of 1600xN.

5.3 Method 3:
In digital signal processing, FFT is a potent tool to extract features. Transforming the time domain into the frequency domain reveals much information about the signal characteristics. Boris et al. [2] used the same method in their work mentioned before. First, we applied FFT transformation on our raw IQ dataset \(C(t)\). Later, the mapping was the same as method 1.

\[
FFT(C(t)) \rightarrow I, Q \quad (16)
\]

Visual representation of the data is in figure 8 and 9.

We used these three different datasets from three different methods to train our deep neural network separately and observed how they perform in the modulation recognition task. Merima et al. [7] worked with these types of the dataset to train their CNN for modulation recognition and wireless interference identification task. In the next section we have discussed the results of the training and prediction by our DNN.

6. Results
As mentioned in previous sections, we have worked on NASA Ka-band dataset to train a DNN to predict the signal modulation, i.e., BPSK or QPSK. We have three different datasets with the dimension of 1600xN. Our neural network structure is given in table 1.
Table 1: DNN structure

<table>
<thead>
<tr>
<th>Layer Type</th>
<th>Parameters</th>
<th>Activation function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense Layer</td>
<td>10 neurons, input dimension = 1600</td>
<td>Leaky ReLU</td>
</tr>
<tr>
<td>Dense Layer</td>
<td>250 neurons, activity-regularizer(0.01)</td>
<td>Leaky ReLU</td>
</tr>
<tr>
<td>Dense Layer</td>
<td>250 neurons</td>
<td>Leaky ReLU</td>
</tr>
<tr>
<td>Dense Layer</td>
<td>1 neuron, output layer</td>
<td>Sigmoid</td>
</tr>
</tbody>
</table>

Using adam optimization training with binary cross-entropy as cost function we trained the DNN with each of the three datasets. We used Keras [12] library with tensorflow backend to build our DNN. To visualize the prediction accuracy we used confusion matrix using sci-kit-learn [13] library. Activation regularizer has been used to address the overfitting problem.

To observe the first dataset performance (raw IQ dataset) we can look at the figures in 10 and 11. After training for 500 epochs the NN cost function converges below 0.05 that is the loss parameter, the accuracy parameter converged at 0.98. If we see the confusion matrix to see the prediction accuracy, for BPSK modulation the model predicted 1002 True Positive(TP), while 38 prediction was False Positive(FP). Similarly, for QPSK modulation the model predicted 1210 True Negative(TN), while 38 results were False Negative(FN).

Fig. 10: Training parameters history of 500 epochs (I,Q dataset)

Fig. 11: Prediction accuracy for modulation recognition (I,Q dataset)

The second dataset (Amplitude and phase dataset) training results have been shown in figure 12. After 500 epochs, the loss has converged at 0.03 and accuracy at 0.99. Modulation prediction in figure 13 for BPSK resulted in 964 TP and 85 FP, where QPSK modulation resulted in 1162 TN and 72 FN.

Fig. 12: Training parameters history of 500 epochs (A,φ dataset)

Fig. 13: Prediction accuracy for modulation recognition (A,φ dataset)

In figure 14, After 500 epochs of training with the FFT I, Q dataset, the loss has converged at 0.06 and accuracy at 0.97. Modulation prediction in figure 15 for BPSK resulted in 948 TP and 24 FP, where QPSK modulation resulted in 1233 TN and 92 FN.

Fig. 14: Training parameters history of 500 epochs (FFT I,Q dataset)

Fig. 15: Prediction accuracy for modulation recognition (FFT I,Q dataset)

From the results, we see that the three different datasets perform differently. For raw I, Q dataset the prediction
precision for BPSK and QPSK is 96%, where with A and $\phi$ dataset gives the precision of 94% for both modulations. FFT I, Q dataset has a high precision rate for QPSK modulation, i.e. 98%, where BPSK modulation precision is low.

7. Conclusion

In this paper, we discussed the different representation of raw I, Q data to train an MLP neural network for modulation recognition. We discussed how the three different datasets trained the model and then how the precision rate varied for each model. For our future work, we are considering the idea of an ensemble algorithm to observe if it increases the precision rate. Also, Convolutional Neural Network (CNN) can be trained to compare the results.

References


