A Hybrid Approach for Signal Modulation Recognition Using Deep Learning Methods

Changjun Fan, Yufeng Wang*, Ming Liu

*The 52nd Research Institute of China Electronic Technology Group Corporation, No. 9, Wenfu Road, Hangzhou 311000, China

*wangyufeng043@cetc.com.cn

Abstract. In this study, a novel signal modulation recognition framework has been proposed for automatically classifying eleven different modulation types with various SNR values. The framework employs both the raw complex-valued I/Q signal and its time-frequency description to represent the radio signal. And, a hybrid deep neural network is presented to recognize different modulation types from the representation data by leveraging the appealing properties of a convolutional neural network (CNN) and a long short-term memory (LSTM) network. Extensive validation of our scheme is performed on a large public dataset by comparing it with three existing methods from literature, and our scheme yields quite promising results in terms of recognition accuracy.

Keywords: Signal Modulation Recognition, Short-time Fourier Transform, Convolutional Neural Network, Long Short-term Memory.

1. Introduction

Wireless communications play an important role in modern society. In a communication system, it is a routine to modulate transmitted signals using carriers in a specific frequency range to make it more suitable for efficient transmission. Being an intermediate phase between signal detection and demodulation, modulation recognition is gaining more and more interest, and it has already been applied to some practical applications involving various civilian or military scenarios. Modulation recognition is an essential step to provide modulation information of signals for further signal demodulation and decoding.

Over the last few years, previous works have suggested the use of various methods to recognize different modulation types. Such methods can be categorized into two general classes: likelihood-based method and feature-based method. The former uses the conditional probability density function based on theoretical theory and assumes there is prior information in a Bayesian sense [1]. However, its computational complexity increases with the number of modulation-related parameters, which makes it hard to obtain the exact likelihood function and makes the classifier inefficient. The latter mainly involves two stages: extracting features for data representation and making the decision by classification. There are usually two kinds of key features, including time-domain features, such as instantaneous phase, amplitude, and frequency, as well as their mean, standard deviation, etc. [2], and transform-domain features, such as Fourier transform, wavelet transform, higher order statistics (HOSs), higher order moments (HOMs), and cyclostationary characteristics [3]. Regarding the
classification techniques, artificial neural networks [4], unsupervised clustering techniques, SVM [5], and decision trees [6] are commonly used. For example, Zhang et al. [7] extracted the HOS feature from the modulated signals and leveraged the I/Q information and HOS to classify different modulation types using deep learning architectures. Daldal et al. [8] extracted the time-frequency information from modulation signals into 2D images and recognized six different modulation types based on them by CNN. In these feature-based methods, recognition performance is dependent on the manually designed features, which need expertise in the field of radio signals. And, the more features we extract, the more processing time is needed for feature engineering and classification.

Nowadays, deep learning has been an active and effective research area in machine learning techniques and pattern recognition. It has achieved big success in the areas of many applications, such as computer vision, speech recognition, and natural language processing. Deep learning is large and deep neural networks made up of layers of nonlinear computing units, and it has the ability to generate effective feature extractors for the classification task. So, it draws much attention and has been employed to fulfill the modulation recognition task in previous related works. O’Shea et al. [9] surveyed emerging applications of deep learning to the radio signal processing domain and provided a public open-source synthetic dataset generated using GNU radio for the purpose of modulation recognition. Based on the above-mentioned dataset, the work [10] probed the adaptation of CNN for recognizing different modulation types and compared its recognition performance with that of the expert cyclic moment features based methods. Moreover, the work [11] proposed to automatically recognize modulation types by deep belief network on the basis of the spectral correlation function as signal representation.

When the modulation recognition related literature is examined, it can be seen that some previous works only used raw I/Q radio signals, while some other studies used the extracted time-frequency information to represent the modulated signals. From the study in paper, we can see that the time-frequency information is one key representation for radio signals, and it plays an important role in the modulation recognition task. Inspired by them, we fuse the raw I/Q data and time-frequency information together to represent the modulated signal. In this way, both time-domain and frequency-domain information are fully considered. As for the learning techniques, we design a hybrid architecture by integrating a CNN with an LSTM. In the architecture, CNN is adopted to extract the spatial features of the signal data, while the LSTM is used to learn the sequential dependence in the time-series radio signals.

2. Modulation recognition and time-frequency analysis

In electronics and telecommunications, modulation is the process of varying one or more properties of a carrier signal, usually a periodic waveform, with a modulating signal that typically contains information to be transmitted. The transmission process can be performed by switching the amplitude, phase, or frequency of the carrier between multiple different values. In this way, it is possible to transmit more information through the transmission of a single carrier. In all kinds of communication systems, such as radio, television, mobile phone, etc., radio signals are modulated prior to transmission. In wireless communication, there are various sources of potential radio interference in the surrounding areas, and each of them differs in behaviors or requirements. So, it is essential for a receiver to infer the modulation type from a received radio signal so as to become more aware of the present type of communication scheme and emitter. This is what modulation recognition does.

Let us simplify the wireless communication system into a model as shown in Fig.1. It is mainly made up of three components, namely a transmitter, a channel, and a receiver. The transmission process can be expressed as in Eq. 1.

\[ r(t) = g(s(t)) * c(t) + n(t) \] (1)

Where \( s(t) \) is a time-series signal standing for source information. It is either a continuous signal or a series of discrete bits, which is usually modulated onto a sinusoidal carrier wave using frequency
modulation, amplitude modulation or phase modulation, etc. The modulation operation can be denoted as a function \( g(\cdot) \). Then, the modulated signal \( g(s(t)) \) is transmitted to the receiver via the communication channel \( c(t) \). In this phase, the signal \( g(s(t)) \) obtains some path loss or constant gain term as well as an additive Gaussian white noise denoted as \( n(t) \). At the receiver end, the observed signal \( r(t) \) is then obtained. The goal of modulation recognition is the inverse of Eq. 1, which needs to obtain the modulation information for estimating the transmitted symbol \( s(t) \) from the observed signal \( r(t) \). In the discrete-time case, sampling the continuous-time signal \( r(t) \) at a certain rate we can get \( r(n) \), \(-\infty < n < +\infty\).

![A simplified communication system model](image)

**Fig. 1** A simplified communication system model

In this study, in addition to raw I/Q signals, we use the time-frequency information of the observed signals to represent the received signals. Time-frequency methods are based on the fact that the corresponding waveform is divided into short-term segments. The time-frequency information is obtained by computing the squared magnitude of the short-time discrete Fourier transform (STFT) of the observed signals. To be specific, a window function is first applied to each equally divided part of the data, and then the Fourier transformation is applied to these parts. The equation used for the discrete-time spectrum analysis is given as in Eq. 2.

\[
R(k, \tau) = \sum_{n=1}^{N} r[n]w[n-\tau]e^{-jnk}
\]  

(2)

Where \( R(k, \tau) \) is the discrete Fourier transform (DFT) coefficient, \( w[n-\tau] \) is the window function with length \( n \), and \( \tau \) is the variable that shifts the window against the \( r[n] \) waveform. In consequence, the linear time-frequency information is calculated as the squared magnitude of STFT given by Eq. 3.

\[
\tilde{R}(k, \tau) = |R(k, \tau)|^2
\]  

(3)

3. Design of deep neural networks

Modulation recognition for \( N \) modulation types can be seen as an \( N \)-class classification problem. To classify modulation types, we design a hybrid model consisting of a CNN and an LSTM, which is called deepCL. The input for deepCL is the 3-channel representation data, including the sampled in-phase and quadrature components of a radio signal and the time-frequency information in the 1-D form. And the output is the corresponding modulation type. The proposed architecture of deepCL is shown in Fig.2.
The first component of deepCL is the 1-dimension convolution operation module based on CNN. CNN can be seen as one specific kind of deep feed-forward artificial neural networks. They are constructed by cascades of convolutional layers. Each of the layers has a number of neurons to carry out the different operations on input data. They extract certain local features from an input data sample in the previous layer by a set of learnable filters and result in the learned feature maps in the next layer. The filters, also called convolutional kernels, are sharing the connection weights between the nodes in the convolutional layers. Sometimes the convolutional layers are followed by a pooling or norm layer to reduce the size of feature maps as well as the computational complexity. Here, we use 4 convolutional layers to extract the spatial features from the input. In every layer, there are 32 convolutional kernels with a dimension of 9 × 1 for each kernel. Furthermore, we apply the rectified linear activation function (Relu) to learn more complex structures in each layer. After being processed by the convolutional layers, the input reaches out to two LSTM layers, and the extracted feature maps by the CNN module become the input to the following LSTM module.

LSTM is one kind of recurrent neural network (RNN), and it is well suitable for processing, classifying, and making predictions based on time-series signal data. A common LSTM unit consists of a memory cell and three gates, i.e., an input gate, an output gate and a forget gate. The three gates adjust the flow of information into and out of the cell and make decisions on what to remember and what to forget over arbitrary time intervals. In this way, it is able to handle the vanishing gradient problem which usually exists in the training process of traditional RNNs. The basic equations that define the forward propagation of a memory cell are shown as follows:

\[
i_t = \sigma(W^i \cdot [h_{t-1}, x_t] + b^i)
\]

\[
f_t = \sigma(W^f \cdot [h_{t-1}, x_t] + b^f)
\]

\[
o_t = \sigma(W^o \cdot [h_{t-1}, x_t] + b^o)
\]

\[
c_t = f_t * c_{t-1} + i_t * \tanh(W^c \cdot [h_{t-1}, x_t] + b^c)
\]

\[
h_t = o_t * \tanh(c_t)
\]

Where \( x_t \) stands for the input vector for the cell at a time \( t \); \( i_t \) stands for the input gate's activation vector, which indicates the degree to which a new value flows into the cell; \( f_t \) stands for the forget gate's activation vector, which indicates the extent to which a value retains in the cell; \( o_t \) stands for the output gate's activation vector, and it indicates the proportion of the value in the cell which is used
to compute the output activation of the unit; And, $c_i$ and $h_i$ denotes the cell state vector and the output of this LSTM unit, respectively. In addition, $W$ and $b$ are the weight matrices and bias vector parameters that need to be learned during training.

The second component of deepCL is made up of two LSTM layers, which are adopted to learn the time-series representation for radio signals. Each of them uses $\tanh(\cdot)$ as the activation function and outputs 64 feature maps for further processing. They are followed by a 'Flatten layer', which converts the spatial dimensions of the feature maps output by the LSTM into 1-D vectors for the convenience to be passed to the following decision layers. Next is a 'Dropout layer', which randomly sets input units to 0 with a certain rate at each step during training time so as to help prevent overfitting. In the end, there is the 'Dense layer' with $\text{soft max}(\cdot)$ as its activation function, and it is responsible for making the decisions by classifying the input signal samples into $N$ classes.

4. Experiments

4.1. Dataset for validation

The proposed approach is evaluated using the RadioML 2016.10a dataset created by O’Shea et al. [9]. It is a synthetic dataset, generated with GNU Radio, made up of 11 modulation methods (8 digital and 2 analog), which is PAM4, WB-FM, BPSK, QPSK, 8PSK, BFSK, 16QAM, 64QAM, CPFSK, AM-DSB, and AM-SSB. These modulation techniques are widely used in practical communication systems and can operate on discrete binary alphabets or continuous alphabets. For each modulation type in the dataset, there are different signal-to-noise ratios (SNR) for use in measuring performance across different signal and noise power scenarios. So, the label for each signal in the dataset includes two parts: the SNR and the modulation mode. Each modulation mode has the same number of signal samples, so does each of the SNRs. And, the SNR of the signals is uniformly distributed from -20 dB to +18 dB, with a step size of 2 dB. To be specific, there are 1000 signals per modulation mode per SNR. Each signal consists of 128 samples, and each sample has separated real and imaginary parts. So, every raw I/Q signal can be expressed using a $2 \times 128$ vector, and the dataset can be denoted as $\text{IQ} \in \mathbb{R}^{128 \times 128}$. In addition to the raw I/Q signals, we extract the time-frequency information for each of them using the frame-based STFT processing. In order to be compatible with the 1-D I/Q signal, we fulfill such a task using the 'stft' function in Scipy [12] by setting the frame length to 16 samples and using a 50%-overlapping Hann window. After being processed by 'stft', we get a 2-D time-frequency image, which is then resized and converted into one complex-valued 1-D vector with a length of 128. Next, we use the 'abs' function in Numpy to get the 1-D modulus information and concatenate it with the raw I/Q signal into a $3 \times 128$ representation denoted as $\text{IQF} \in \mathbb{R}^{3 \times 128}$. To facilitate the comparisons in the following experiments, we also create a dataset made of only time-frequency information and denote it as $\text{OF} \in \mathbb{R}^{6 \times 128}$.

4.2. Experimental results and analysis

For the sake of convenience to evaluation, the following preparatory work is carried out to set up the experiments. First, we create two new models by trimming the deepCL architecture, which we refer to as deepC and deepL, respectively. The former is obtained by removing the LSTM component from deepCL, and the latter is gotten by removing the CNN component from deepCL. Then, the three models are applied to the datasets of IQ, OF, and IQF, and the corresponding methods are denoted as deepCL_IQF, deepC_IQF, deepL_IQF, deepCL_OF and deepCL_IQ, etc. Moreover, we obtain three methods from previous related works, which are CNN-IQFOC and LSTM-IQFOC from the work [8] and CNNR from the work, for comparison. To get the dataset ready, 70% of the samples are randomly selected as training data for each modulation mode with each SNR, and 20% of the samples are
selected as validation data, the remaining data as test data. In addition, we carry out the experiments using Nvidia GTX 1080 GPU on Ubuntu 18.04.

First, we compare the recognition performance among deepCL, deepC, and deepL based on the IQF dataset, and Fig.3 shows the results for us. From Fig.3 we can see that the recognition accuracy of deepCL_IQF is higher than that of both deepC_IQF and deepL_IQF when SNR is above -6 dB. And, the recognition accuracy of deepCL_IQF has exceeded that of deepC_IQF about 6% when the performance curve becomes steady with SNR above 0 db. During the experiment, we also find out that the number of deepL_IQF's parameters is about 2 times more than that of deepC_IQF's parameters. Still, the recognition performance of the latter is better than that of the former when SNR is above -2 dB. However, once the SNR decreases to -6 dB, deepL_IQF seems to perform slightly better than the others. In addition, by applying deepCL to IQ and OF datasets, we find out that the corresponding deepCL_IQ has a good performance, while deepCL_OF performs comparatively poor. However, it's worth noting that deepCL_IQF achieves the best performance among the three methods of deepCL_IQ, deepCL_OF, and deepCL_IQF, which use different kinds of data to represent the signals. This indicates that our hybrid model takes full advantage of both the merits of the CNN and LSTM and the time-frequency information we introduce provides supplementary representation ability to the I/Q signal.

![Fig.3 Recognition accuracy comparison between different models and signal representation](image)

Then, the recognition performance of our deepCL_IQF is compared with the CNN-IQFOC, LSTM-IQFOC, and CNNR from the previous studies. The experimental results in Fig.4 show us that our scheme achieves the best performance among 4 different methods when the SNR is above -2dB, and its recognition accuracy is nearly 5% higher than that of the CNN-IQFOC and LSTM-IQFOC. For all the scenarios with different SNR values, LSTM-IQFOC performs a little bit better than CNN-IQFOC, while they both have a much higher recognition accuracy than CNNR. When the SNR decreases from 0 dB to -14 dB, all the four methods have a noticeable decline in the performance. And, CNN-IQFOC, LSTM-IQFOC, and deepCL_IQF have a similar performance while the SNR is below -12 dB, which indicates that they are all not good at making the right predictions for the noisy signals. Even though the recognition performance for CNN-IQFOC, LSTM-IQFOC, and deepCL_IQF roughly remain at the same level when the value of SNR is between -12 dB and -2 dB, it’s intriguing to see that the performance of LSTM-IQFOC is slightly better than deepCL_IQF. However, when the SNR increases up to -2 dB, even our deepCL_IQ (in Fig.3) has a higher recognition accuracy than CNN-IQFOC and LSTM-IQFOC.
Fig. 4 Modulation recognition accuracy compared with previous methods

All the results from the above experiments indicate that the proposed signal representation and hybrid network architecture have a significant impact on the recognition performance and they can improve the recognition accuracy significantly, especially when the value of SNR is high.

5. Conclusion
In this study, we proposed to combine the raw I/Q data and its time-frequency information to represent radio signals. And, we presented a hybrid deep neural network architecture to leverage the appealing properties of a CNN and an LSTM. The modulation recognition performance for different signal representation was compared, and it came to a conclusion that the proposed IQF representation is less sensitive to noise and tends to obtain good classification performance by introducing the STFT information to the raw radio signals. Furthermore, we can see from the experimental results that our hybrid architecture, deepCL, do much better than deepC or deepL, which uses only a CNN or an LSTM. The proposed method can not only extract the spatial features of the signals but also learn the sequential dependence in the time-series signals effectively. In addition, we made a comparison between our architecture and three methods from previous related works and concluded that our scheme can achieve better recognition performance by means of the proposed hybrid architecture and effective IQF representation.

References
[1] Wei W, Mendel J M 2000 Maximum-likelihood classification for digital amplitude-phase modulations. *IEEE J. transactions on Communications* 48(2) pp189-193
[2] Aisbett J 1987 Automatic modulation recognition using time domain parameters *J. Signal processing* 13(3) pp323-328
[3] Wong M D, Nandi A K 2001 Automatic digital modulation recognition using spectral and statistical features with multi-layer perceptrons. *J In Proceedings of the sixth international symposium on signal processing and its applications* (Cat. No. 01EX467) 2 pp390-393. IEEE
[4] Yaqin Z, Guanghui R, Xuexia W, Zhilu W and Xuemai G 2003 Automatic digital modulation recognition using artificial neural networks *International Conference on Neural Networks and Signal Processing* 1 pp257-260
[5] Shiping L, Fangchao C and Long W 2012 Modulation recognition algorithm of digital signal based on support vector machine *The 24th Chinese Control and Decision Conference* (CCDC). IEEE
[6] Tong L, Yanan G and Yun L 2017 Research on modulation recognition with ensemble learning. *EURASIP J.Journal on Wireless Communications and Networking* 1 179
[7] Meng Z, Yuan Z, Zidong H and Yi G 2018 Automatic modulation recognition using deep
learning architectures *Proc. 19th IEEE Int. Workshop Signal Process. Adv Wireless Commun (SPAWC)* pp1–5

[8] Daldal N, Cömert Z and Polat K 2020 Automatic determination of digital modulation types with different noises using Convolutional Neural Network based on time-frequency information. *J. Applied Soft Computing* p105834

[9] O’Shea T J and West N 2016 Radio machine learning dataset generation with GNU radio. *Proc. GNU Radio Conf.* 1 pp1–6

[10] O’Shea T J, Corgan J and Clancy T C 2016 Convolutional radio modulation recognition networks *In International conference on engineering applications of neural networks* pp213-226. Springer, Cham

[11] Mendis G J, Wei J and Madanayake A 2016 Deep learning-based automated modulation classification for cognitive radio *Proc. IEEE Int. Conf. Commun. Syst.* (ICCS), Shenzhen, China pp1–6

[12] Johansson 2018 Numerical Python Scientific Computing and Data Science Applications with Numpy SciPy and Matplotlib *Apress*