DEEP NEURAL NETWORK ARCHITECTURES FOR MODULATION CLASSIFICATION

A Thesis

Submitted to the Faculty

of

Purdue University

by

Xiaoyu Liu

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2018

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVAL

Prof. Aly El Gamal, Chair

School of Electrical and Computer Engineering

Prof. David Love

School of Electrical and Computer Engineering

Prof. Jeffrey Siskind

School of Electrical and Computer Engineering

Approved by:

Venkataramanan Balakrishnan

Head of the School Graduate Program

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. Aly El Gamal for the continuous support of my M.Sc. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of the research and writing of this thesis. I am really appreciative of the opportunities in both industry and academia that he provided for me, and the kind recommendations from him.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. David Love and Prof. Jeffrey Siskind not only for their insightful comments, but also for the hard questions which incentivized me to widen my research from various perspectives.

I also would like to thank my labmate, Diyu Yang, for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we had in the last year.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of research. This accomplishment would not have been possible without them. Thank you.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

SDR	software define radio
AMC	automatic modulation recognition
SNR	signal noise ratio
LB	likelihood-based
FB	feature-based
LRT	likelihood ratio test
ALRT	average likelihood ratio test
GLRT	generalized likelihood ratio test
AWGN	additive white Gaussian noise
HLRT	hybrid likelihood ratio test
KNN	K-nearest neighborhood
WT	wavelet transform
HWT	Haar wavelet transform
DWT	digital wavelet transform
PSD	power spectral density
HOM	higher order moments
HOC	higher order cumulants
SVM	support vector machine
MLP	multi-layer perceptron
RBF	radial basis function
DNN	deep neural network
CNN	convolutional neural network
DenseNet	densely connected neural network
LSTM	long short-term memory

ResNet	residual neural network
CLDNN	convolutional LSTM dense neural network
ILSVRC	ImageNet large scale visual recognition challenge

ABSTRACT

M.S., Purdue University, May 2018. Deep Neural Network Architectures for Modulation Classification. Major Professor: Aly El Gamal.

This thesis investigates the value of employing deep learning for the task of wireless signal modulation recognition. Recently in deep learning research on AMC, a framework has been introduced by generating a dataset using GNU radio that mimics the imperfections in a real wireless channel, and uses 10 different modulation types. Further, a CNN architecture was developed and shown to deliver performance that exceeds that of expert-based approaches. Here, we follow the framework of O'shea [1] and find deep neural network architectures that deliver higher accuracy than the state of the art. We tested the architecture of O'shea [1] and found it to achieve an accuracy of approximately 75% of correctly recognizing the modulation type. We first tune the CNN architecture and find a design with four convolutional layers and two dense layers that gives an accuracy of approximately 83.8% at high SNR. We then develop architectures based on the recently introduced ideas of Residual Networks (ResNet) and Densely Connected Network (DenseNet) to achieve high SNR accuracies of approximately 83% and 86.6%, respectively. We also introduce a CLDNN to achieve an accuracy of approximately 88.5% at high SNR. To improve the classification accuracy of QAM, we calculate the high order cumulants of QAM16 and QAM64 as the expert feature and improve the total accuracy to approximately 90%. Finally, by preprocessing the input and send them into a LSTM model, we improve all classification success rates to 100% except the WBFM which is 46%. The average modulation classification accuracy got a improvement of roughly 22% in this thesis.

1. INTRODUCTION

1.1 Motivation

Wireless communication plays an important role in modern communication. Modulation classification, as an intermediate process between signal detection and demodulation, is therefore attracting attention. Modulation recognition finds application in commercial areas such as space communication and cellular telecommunication in the form of Software Defined Radios (SDR). SDR uses blind modulation recognition schemes to reconfigure the system, reducing the overhead by increasing transmission efficiency. Furthermore, AMC serves an important role in the information context of a military field. The spectrum of transmitted signals spans a large range and the format of the modulation algorithm varies according to the carrier frequency. The detector needs to distinguish the source, property and content correctly to make the right processing decision without much prior information. Under such conditions, advanced automatic signal processing and demodulation techniques are required as a major task of intelligent communication systems. The modulation recognition system essentially consists of three steps: signal preprocessing, feature extraction and selection of modulation algorithm. The preprocessing may include estimating SNR and symbol period, noise reduction and symbol synchronization. Deep learning algorithms have performed outstanding capabilities in images and audio feature extraction in particular and supervised learning in general, so it naturally comes as a strong candidate for the modulation classification task. To give a comprehensive understanding of AMC using deep learning algorithms, this project applies several state-of-art neural network architectures on simulated signals to achieve high classification accuracy.

1.2 Background

Over the past few decades, wireless communication techniques have been continuously evolving with the development of modulation methods. Communication signals travel in space with different frequencies and modulation types. A modulation classification module in a receiver should be able to recognize the received signals modulation type with no or minimum prior knowledge. In adaptive modulation systems, the demodulators can estimate the parameters used by senders from time to time. There are two general classes of recognition algorithms: likelihood-based (LB) and feature-based (FB). The parameters of interest could be the recognition time and classification accuracy.

A general expression of the received baseband complex envelop could be formulated as

$$r\left(t\right) = \ s(t;\boldsymbol{u}_{\mathbf{i}})\ \ + n\left(t\right), \tag{1.1}$$

where

$$s(t;\boldsymbol{u}_{\mathbf{i}})\ =\ a_i e^{j2\pi\triangle ft} e^{j\theta} \textstyle\sum_{k=1}^{K} e^{j\phi_k} s_k^{(i)} g(t-(k-1)T-\varepsilon T),\ 0 \leq t \leq KT \tag{1.2}$$

is the noise-free baseband complex envelope of the received signal. In (1.2), a_i is the unknown signal amplitude, f is the carrier frequency offset, θ is the time-invariant carrier frequency introduced by the propagation delay, ϕ_k is the phase jitter, s_k denotes the vector of complex symbols taken from the i^{th} modulation format, T represents the symbol period, ε is the normalized epoch for time offset between the transmitter and signal receiver, $g(t) = P_{pulse}(t) \otimes h(t)$ is the composite effect of the residual channel with $h(t)$ as the channel impulse response and $\otimes$ as the convolution. $\boldsymbol{u}_{\mathbf{i}} = \{a,\ \theta,\ \varepsilon,\ h(t),\ \{\varphi_n\}_{n=0}^{N-1},\ \{s^{k,i}\}_{k=1}^{M_i},\ \omega_c\}$ is used as a multidimensional vector that includes deterministic unknown signal or channel parameters such as the carrier frequency offset for the i^{th} modulation type.

1.2.1 Likelihood-Based Methods

The LB-AMC has been studied by many researchers based on the hypothesis testing method. It uses the probability density function of the observed wave conditioned on the intercepted signal to estimate the likelihood of each possible hypothesis. The optimal threshold is set to minimize the classification error in a Bayesian sense. Therefore, it is also called the likelihood ratio test (LRT), because it's a ratio between two likelihood functions. The steps in the LB model are shown in Figure 1.1. The receiver measures the observed value of the input signal, then calculates the likelihood value under each modulation hypothesis H. So the likelihood is given by

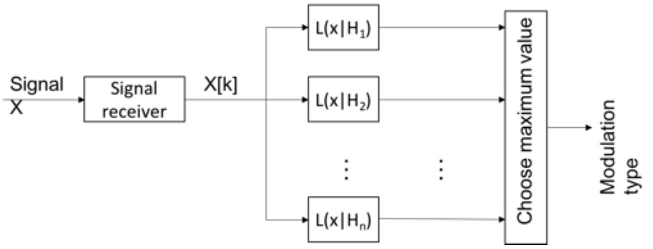


Fig. 1.1. Likelihood-based modulation classification diagram

$$\Lambda_A^{(i)}[r(t)] \;=\; \int \Lambda[r(t)|v_i, H_i]p(v_i|H_i)dv_i, \tag{1.3}$$

where $\Lambda[r(t)|v_i, H_i]$ is the conditional likelihood function given H_i and unknown vector v_i for the i^{th} modulation scheme, $p(v_i|H_i)$ is the prior probability density function. The estimated modulation algorithm is finally decided by the probability density functions. The average likelihood ratio test(ALRT) algorithm proposed by Kim in 1988 [2], which successfully distinguished between BPSK and QPSK, is the first LB algorithm based on Bayesian theory. The authors in [2] assumed that signal parameters such as SNR, the symbol rate and carrier frequency are available for the recognizer. These parameters are regarded as random variables and their probability density functions are evenly calculated. The log-likelihood ratio was used to estimate the modulation scheme, that is to say, the number of levels, M, of the M-PSK signals.

The condition likelihood function is derived with a baseband complex AWGN when all necessary parameters are perfectly known. It is given by

$$\Lambda[r(t)|v_i, H_i] \ = exp\left[2N_0^{-1}Re\left\{\int_0^{KT} r(t)s*(t;u_i)dt\right\} - N_0^{-1}\int_0^{KT}|s(t;u_i)|^2dt\right], \tag{1.4}$$

where N_0^{-1} is the two-sided power spectral density and $*$ is the complex conjugate. Kim et al. [2] also did a comparison between three different classifiers which are a phase-based classifier, a square-law based classifier and a quasi-log-likelihood ratio classifier. The last one turned out to perform significantly better than the others. The ALRT algorithm was further developed by Sapiano [3] and Hong [4] later. While ALRT's requirement of the full knowledge of prior information and multidimensional integration renders itself impractical, Panagiotou et al [5] and Lay et al [6] treated the unknown quantities as unknown deterministics and the algorithm is named GLRT since it uses maximum likelihood for probability density function and feature estimation. The Generalized LRT treats parameters of interest as determinstics, so the likelihood function conditioned on H_i is given by

$$\Lambda_G^{(i)}[r(t)] \ = \underset{v_i}{max}\Lambda[r(t)|v_i, H_i]. \tag{1.5}$$

The best performance of this algorithm was achieved by UMP test [7]. For an AWGN channel the likelihood function is given by

$$\Lambda_G^{(i)}[r(t)] \ = \underset{\theta}{max}\left\{\textstyle\sum_{k=1}^{K}\underset{s_K^{(i)}}{max}(Re[s_K^{(i)*}r_k e^{-j\theta}] - 2^{-1}\sqrt{S}T|s_k^{(i)}|^2)\right\}. \tag{1.6}$$

The generalized likelihood ratio test (GLRT) outperforms ALRT in terms of exponential functions and the knowledge of noise power but suffers from nested signal constellations. Panagiotou et al [5] pointed out that it gets the same likelihood function values for BPSK, QPSK, QAM-16 and QAM-64. HLRT [8] was therefore introduced as a combination of ALRT and GLRT. The hybrid model solves the multidimensional integration problem in ALRT and the nested constellations problem in

GLRT by averaging unknown symbols. The likelihood function of this algorithm is given by

$$\Lambda_H^{(i)}[r(t)] \; = \underset{v_i}{max} \int \Lambda[r(t)|v_{i_1}, v_{i_2}, H_i] p(v_{i_2}|H_i) dv_{i_2}, \tag{1.7}$$

where $\boldsymbol{v}_{\mathbf{i_1}}$ and $\boldsymbol{v}_{\mathbf{i_2}}$ are unknown deterministic vectors, $\boldsymbol{v}_{\mathbf{i}} = [\mathbf{v_{i_1}} \; \mathbf{v_{i_2}}]$ denotes unknown vectors. When the distribution of $\boldsymbol{u}_{\mathbf{i}}$ is unknown in the hybrid likelihood ratio test (HLRT) algorithm, the maximum likelihood estimations of the unknown parameters are used as substitutions in log likelihood functions. By substituting the likelihood function for an AWGN channel model into (1.1), the function is given by

$$\Lambda_H^{(i)}[r(t)] \; = \underset{\theta}{max} \left\{ \prod_{k=1}^{K} E_{S_k^{(i)}} \left\{ exp \left[2\sqrt{S} N_0^{-1} Re \left[s_k^{(i)*} r_k e^{-j\theta} \right] - STN_0^{-1} |s_k^{(i)}|^2 \right] \right\} \right\}, \tag{1.8}$$

with $\boldsymbol{u}_{\mathbf{i}} = \left[\theta \; S \; \left\{ s_k^{(i)} \right\}_{k=1}^{K} \right]$ where θ is an unknown phase shift obtained by two-step processing. Since the maximum likelihood estimations are functions of $s(t)$, all symbol sequences with length of K would be taken into account. The complexity is therefore in the order of $O\left(NM_m^K\right)$ when there are m types of modulation hypotheses. Lay et al [9] applied per-survivor processing technique, a technique for estimating data sequence and unknown signal parameters which exhibits memory, in an inter symbol interference environment. In [10], a uniform linear array was used to better classify BPSK and QPSK signals at low SNR based on the HLRT algorithm, with $\boldsymbol{v}_{\mathbf{i}} = \left[\theta \; \left\{ s_k^{(i)} \right\}_{k=1}^{K} \right]$. Dobre [11] built classifiers based on the HLRT in flat block fading channels, with $\boldsymbol{v}_{\mathbf{i}} = \left[\alpha \; \varphi \; \left\{ s_k^{(i)} \right\}_{k=1}^{K} \; N_0 \right]$. The decision threshold was set to one and the likelihood functions were computed by averaging over the data symbols.

Although W. Wen [12] proved that ALRT is the optimal classification algorithm under Bayesian rule, the unknown variables and the computation increase significantly in complex signal scenarios. Quasi likelihood tests were introduced to solve the problem including quasi-ALRT [13] and quasi-HLRT [14] which are said to be suboptimal structures. The study on qALRT originated in [2] where only BPSK and QPSK were considered in derivation. [8] generalized the study cases to M-PSK as well as comprehensive simulations while [15] extended the qALRT algorithm to the

M-QAM signals. J. A. Sills et al. [14], A. Polydoros et al. [13], also used similar methods to get approximate LRT. They used the likelihood ratio functions that best match signals from filters to classify digital signals, therefore reducing the number of unknown variables and computational complexity. qALRT based classifiers introduce timing offset to transform the classifiers into asynchronous ones. The likelihood function is given by

$$\Lambda_A^{(i)}[r(t)] \ \approx D^{-1}\textstyle\sum_{d=0}^{D-1}\Lambda[r(t)]|\varepsilon_d, H_i], \tag{1.9}$$

where D is the number of timing offset quantized levels, ε_d equals d/D, $d = 0, ..., D-1$. As $D \to \infty$, the summation in (1.9) converges to the integral making the approximation improve. However, the larger D is, the higher complexity is resulted as more terms are introduced in (1.9). Dobre et al. [11,16,17] developed the qHLRT algorithm to estimate the unknown noise variance of linear digital modulations in block fading, with $\boldsymbol{v}_{\mathbf{i}} = \left[\alpha \ \varphi \ \left\{s_k^{(i)}\right\}_{k=1}^{K} \ N_0\right]$. [11] proposed a modulation classification classifer for multi-antenna with unknown carrier phase offset. It also provided simulations by generating normalized constellations for QAM-16, QAM-32 and QAM-64 which achieved a reasonable classification accuracy improvement. [18] proposed a similarity measure from the likelihood ratio method, known as the correntropy coefficient, to overcome the high computational cost in preprocessing. Binary modulation experiments reach a 97% success rate at SNR of 5dB.

LB methods are developed on complete theoretical basis, therefore derive the theoretical curve of the recognition performance and guarantees optimal classification results with minimum Bayesian cost. So it provides an upper bound or works as a benchmark for theoretical performance that can verify the performance of other recognition methods. Besides, by considering noise when building tested statistical models, LB presents outstanding recognition capability in low SNR scenarios. The algorithm can also be further improved for non-perfect channels according to the integrity of the channel information. However, the weakness of the LB approach lies in its computational complexity which may make the classifier impractical. When the number of unknown variables increases, it is hard to find the exact likelihood function.

The LRT approximation likelihood function, so-call quasi-ALRT algorithm, however, will decrease the classification accuracy due to the simplification. LB methods have therefore a lack of applicability because the parameters of the likelihood function are derived for specific signals under certain conditions, so it only suits specific modulation recognition scenarios. Besides, if the assumption of prior information is not satisfied, the LB approach performance would decline sharply when the parameters are not estimated correctly or the built model does not match the real channel characteristics.

1.2.2 Feature-Based Method

A properly designed FB algorithm can show the same performance as the LB algorithm but suffers from less computation complexity. The FB method usually includes two stages: extracting features for data representation and the decision making, i.e. classifiers. The general process of FB is illustrated in Figure 1.2. The key features can be categorized as time domain features including instantaneous amplitude, phase and frequency [19] [20] [21], transform domain features such as wavelet transform or Fourier transform of the signals [22]- [23], higher order moments(HOMs) and higher order cumulants(HOCs) [24]. The fuzzy logic [25] and constellation shape features [26] [27] are also employed for AMC. The classifiers or pattern recognition methods include artificial neural networks [28], unsupervised clustering techniques, SVM [29] and decision tree [30]. DeSinio [21] derived features from the envelope of the signal and from the spectra of the signals and the signal quadrupled for BPSK and QPSK. Ghani [31] did a classification performance comparison between K-nearest

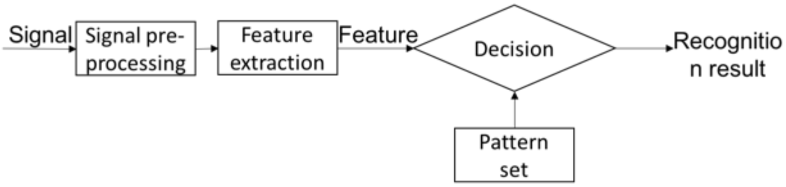


Fig. 1.2. Feature-based modulation classification diagram

neighbor (KNN) and ANN using power spectrum density for discriminating AM, FM, ASK, etc. In 1995, Azzouz and Nandi [19] [32] used instantaneous carrier frequency, phase and amplitude as key features and ANN as classifier, and conducted the recognition of analogue and digital signal schemes, which was considered as a new start of FB methods. Their simulation results show that the overall success rate is over 96% at the SNR of 15 dB using an ANN algorithm. It is indicated in [19] that the amplitude in 2-ASK changes in two levels which equal in magnitude but oppose in sign. So the variance of the absolute value of the normalized amplitude contains no information, whereas the same function for 4-ASK contains information. A threshold is set in the decision tree for that distinguishing statistic. The maximum of the discrete Fourier transform of the instantaneous amplitude is calculated to discriminate FSK and PSK/ASK, as for the former the amplitude has information whereas it does not have for the latter two. M-PSK and ASK are distinguished according to the variance of the absolute normalized phase as ASK does not have phase information. The classifier is again chosen to be a binary decision tree. Zhinan [20] derived the instantaneous analytical signal amplitude from Hilbert transform then used it to obtain clustering centers. Given the Hilbert transform $\widehat{r}(t)$ of the received signal $r(t)$, the instantaneous amplitude, frequency and phase are given by

$$a(t) = |z(t)| = \sqrt{r^2(t) + \widehat{r}^2(t)}, \tag{1.10}$$

$$\varphi(t) = unwrap\,(angle\,(z\,(t))) - 2\mathrm{tf_c t}, \tag{1.11}$$

$$f_N = \frac{1}{2\pi}\frac{d\,(arg\,(z\,(t)))}{dt}. \tag{1.12}$$

The computer simulations showed that M-QAM recognition performance increases as the SNR increases. Hsue et al [33] used zero-crossing interval which is a measure of instantaneous frequency. By utilizing the character of zero-crossing interval that it is a staircase function for FSK but a constant for PSK and unmodulated waveform, AMC becomes a two hypothesis testing problem. The Gaussian assumption is simplified to the feature comparison using LRT. K.C. Ho et al [22] [34] used wavelet transform (WT) to localize the change of instantaneous frequency, amplitude and phase. For PSK, the Haar wavelet transform (HWT) is a constant while HWT becomes staircase functions for FSK and QAM because of the frequency and amplitude changes. FSK can be distinguished from PSK and QAM according to the variance of the HWT magnitude with amplitude normalization. The HWT magnitude without amplitude normalization could be used for discrimination between QAM and PSK. In digital wavelet transform (DWT), the intercepted signals are divided into two bands recursively. By this decomposition method, the resolution in frequency domain increases, making the decision making classifier easier [35]. Both works on analogue [36] and digital signals [37] employed power spectral density (PSD) for classifications. The maximum value of PSD of normalized centered instantaneous amplitude derived from the Fourier transform is given by

$$\gamma_{max} = \frac{max|DFT\left(a_{cn}\left(n\right)\right)|^2}{N_s}. \tag{1.13}$$

The γ_{max} represents the amplitude variance, therefore it is employed to distinguish between AM and FM, M-QAM and PSK. [23] also used PSD as well as the derivation of instantaneous amplitude, frequency and phase to derive key features. A threshold was decided for the above features. Simulations show that the classification accuracy is higher than 94% when the SNR is 10dB. High order statistics [24] are composed of HOCs and HOMs and are used for M-PSK, QAM and FSK classifications. The HOM of the intercepted signal is expressed as

$$M_{p+q,p} = E\left[x\left(n\right)^p\left(x\left(n\right)^*\right)^q\right], \tag{1.14}$$

where $x(n)$ is the input signal. [38] used this method to discriminate between QPSK and QAM-16. The decision is made depending on the correlation between the theoretical value and estimated one. The cumulant is defined by $C_{n,q}(0_{n-1})$ representing the n^{th} order/q-conjugate cumulant of the output. By combining more than one HOM, an example of the HOC is given by

$$C_{42} = cum[x(n)\ x(n)\ x(n)\ x(n)]* = M_{41} - 3M_{20}M_{21}. \tag{1.15}$$

Swami et al. [39] used $C_{4,2}$ for ASK, the magnitude of $C_{4,0}$ for PSK and $C_{4,2}$ for QAM. The decision is made to minimize the probability of error. Simulation results in [40] show that maximum likelihood modulation classification produces best results but there is misclassification between QAM-16 and QAM-64 when using the 4^{th} order cumulants. The 6^{th} order cumulant is applied and exhibits large gap between QAM-16 and QAM-64. Since the constellation map characterizes the PSK and QAM signals, Pedzisz et al. [26] transformed the phase-amplitude distributions to one dimensional distributions for discrimination. Based on the information contained in the location of different QAM signals, Gulati et al. [27] proposed classifiers calculating the Euclidean distances between constellation points and studied the effect of noise and carrier frequency offset on success rate. SVM achieves the classification by finding the maximum separation between two classes. RBF and polynomial functions are usually used as kernels that can map the input to feature domains. For multiple class problems, binary SVM is employed. [29] used SVM to solve the multiple classification task by first classifying one class against other classes, then finding a second to be classified against the remaining others, and so on. [30] used a decision tree in AMC to automatically recognize QAM and OFDM. The basic idea of the decision tree is to use a threshold to separate the hypotheses.

We note that FB methods outperform LB methods in terms of preprocessing and the generality. It is based on a simple theory and the performance remains robust even with little prior knowledge or low preprocessing accuracy. But it is vulnerable to noise and non-ideal channel conditions.

1.2.3 ANN

The artificial neural network (ANN) has succeeded in many research areas and applications such as pattern recognition [32] and signal processing [41]. Different kinds of neural networks have been implemented on the second step of feature based pattern recognition, including probabilistic neural networks and the support vector machine. Single multi-layer perceptrons (MLP) have been wildly used as classifiers as reported by L. Mingquan et al. [42] and Mobasseri et al. [43]. Others also suggested using cascaded MLP in ANN [19], in which the output of the previous layers are fed into latter layers as input. Given the same input features, the MLP ANN outperforms the decision tree method. Unlike LB and FB approaches, where the threshold for decision should be chosen manually, the threshold in neural networks could be decided automatically and adaptively. On the other hand, as many decision-theoretic algorithms presented, the probability of a correct decision on the modulation scheme depends on the sequence of the extracted key features. As can be seen that a different order of key feature application results in different success rates for the modulation type at the same SNR. The ANN algorithms deal with this uncertainty by considering all features simultaneously, so that the probability of the correct decision becomes stable.

Sehier et al. [28] suggested a hierarchical neural network with backpropagation training in 1993. An ANN generally includes three steps (see Figure 1.3):

1. Preprocessing of the input signal which is different from the first step in traditional signal processing. The preprocessing step in ANN extracts key features from an input segment.

2. Training phase learns features and adjusts parameters in classifiers.

3. Test phase evaluates the classification performance.

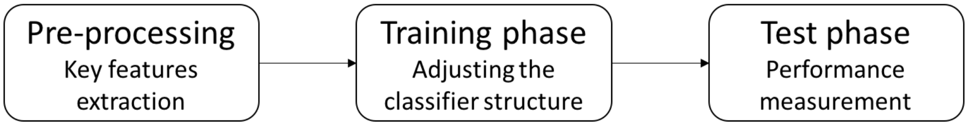


Fig. 1.3. ANN algorithms diagram

During the training process, parameters in the architecture are modified in the direction that minimizes the difference between predicted labels and true labels using the backpropagation algorithm. Sehier et al. [28] also analyzed the the performance of other algorithms such as the binary decision tree and KNN. L. Mingquan et al. [44]

utilized the cyclic spectral features of signals to build a novel MLP-based neural network that can efficiently distinguish modulation types such as AM, FM, ASK and FSK. Mingquanet al. [42] further improved this technique by extracting the instantaneous frequency and occupied bandwidth features. Nandi and Azzouz [45] simulated different types of modulated signals corrupted by a band-limited Gaussian noise sequence to measure the ANN classification performance. The experiments were carried out for ASK, PSK and FSK. They found that their ANN approach reached success rates that are larger than 98% when the SNR is larger than 10dB for both analogue and digitally modulated signals. Their algorithms inspired a bunch of commercial products. An example application for 4G software radio wireless was illustrated in networks [46].

Recently ANN has been studied and improved to present outstanding performance in classification with the development of big data and computation ability. A deeper neural network outperforms traditional ANN by learning features from multilevel nonlinear operations. The concept of DNN was firstly proposed by Hinton [47] in 2006, which refers to the machine learning process of obtaining a multilevel deep neural network by training sample data. Traditional ANNs randomly initialize the weights and the bias in the neural network usually leads to a local minimum value. Hinton et al. solved this problem by using an unsupervised the pre-training method for the weights initialization.

DNN is generally categorized as feed-forward deep networks, feed-back deep networks and bi-directional deep networks. Feed-forward deep networks typically include MLP [19] and CNN [48, 49]. CNN is composed of multiple convolutional layers and each layer contains a convolutional function, a nonlinear transformation and down sampling. Convolutional kernels detect the specific features across the whole input image or signal and achieve the weight sharing, which significantly reduces the computation complexity. Further details on CNN would be introduced in Section 3. The deconvolutional network [50] and hierarchical sparse coding [51] are two examples of feed-back deep networks. The basic idea behind feed-back architectures resembles the convolutional neural network [48], but they differ in terms of implementation. The filters recompose the input signals based on convolutional features using either a convolution matrix or matrix multiplication. The training of bi-directional networks is a combination of feed-forward and feed-back training. A greedy algorithm is employed in the pre-training of each single layer. The input signal I_L and weights W are used to produce I_{L+1} for the next layer, while I_{L+1} and the same weights W are calculated to recompose the signal $I_L^{'}$ mapping to the input layer. Each layer is trained during the iteration of reducing the difference between I_L and $I_L^{'}$. The weights in the whole network are fine tuned according to the feed-back error.

Advanced DNN architectures are largely applied in image recognition tasks and show high success rates in image recognition challenges such as ILSVRC. The DNN model developed by Krizhevsky et al. [52] was the first CNN model application that ranks first at image classification and object detection tasks in ILSVRC-2012. The error of their algorithm was among the top-5, and was 15.3%, which was much lower than the second-best error rate of 26.2%.

Unlike the ANN used in the traditional AMC problem, the deep neural network extracts the features inside its structure, leaving little preprocessing work for the receiver. Traditional AMC algorithms including FB and LB methods were proposed and tested on theoretical mathematical models. In this thesis, we use simulated data as training and testing samples, and the data generation is introduced in Section 2. This thesis also proposes different blind modulation classifiers by applying different state of the art deep neural network architectures as discussed in Section 3. The success rate comparison, analysis and suggestions for future research directions are given in Section 4 and Section 5, respectively.

2. EXPERIMENTAL SETUP

2.1 Dataset Generation

Previous studies of the modulation recognition problems are mainly based on mathematical models, simulation works have also been conducted but limited to only one category of the signal such as only digital signal modulations. Previous studies have also been limited to distinguishing between similar modulation types and a smaller number (2-4), here we have 10. This thesis uses the simulated modulated signal generated in GNU radio [53] with the channel model blocks [54]. A high level framework of the data generation is shown in figure 3, where the logical modules will be explained successively.

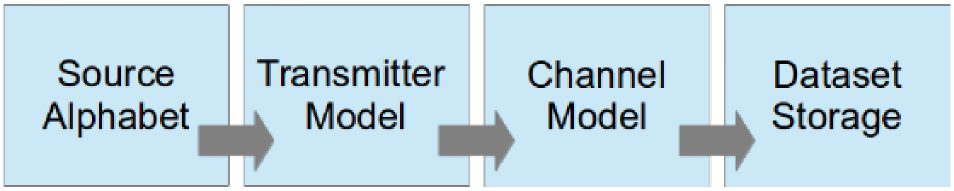


Fig. 2.1. A frame of data generation

2.1.1 Source Alphabet

Two types of data sources are selected for the signal modulation. Voice signals are chosen as continuous signals for analog modulations. The sound from the first Episode of Serial on the podcast which includes some off times is used in this case. For digital modulations, the data is derived from the entire Gutenberg works of Shakespeare in ASCII, and then whitened by randomizers to ensure that bits are equiprobable. The two data sources are later applied to all modems.

2.1.2 Transmitter Model

We choose 10 widely used modulations in wireless communication systems: 2 analog and 8 digital modulation types. Digital modulations include BPSK, QPSK, 8PSK, 16QAM, 64QAM, BFSK, CPFSK, PAM4 and analog modulations consist of WBFM and AM-DSB. Digital signals are modulated at a rate of 8 samples per symbol. The voltage level time series of digital signals are projected onto sine and cosine functions and then modulated through manipulating the amplitude, phase or frequency. The phase mapping of the QPSK, for example, is given by

$$s(t_i) = e^{j2\mathrm{f_c}\mathrm{t}+\frac{2\mathrm{c_i}+1}{4}\pi},\ c_i \in 0, 1, 2, 3. \tag{2.1}$$

PSK, QAM and PAM are modulated using the transmitter module followed by an interpolating pulse shaping filter to band-limit the signal. A root-raised cosine filter was chosen with an excess bandwidth of 0.35 for all signals. The remaining modulations are generated by the GNU radio hierarchical blocks.

2.1.3 Channel Model

In real systems, there are a number of factors that may affect the transmitted signals. The physical environmental noises from industrial sparks, electric switches and the temperature can lead to temporal shifting. The thermal noises caused by semiconductors that are different from the transmitter or the cosmic noise from astronomical radiation can result in white Gaussion noise which can be measured by SNR. Multipath fading occurs when a transmitted signal divides and takes more than one path to a receiver and some of the signals arrive with varying amplitude or phase, resulting in a weak or fading signal. These random processes are simulated using the GNU Radio Dynamic Channel Model hierarchical blocks. The models for generating noises include:

- Sample rate offset model: varies sample rate offset with respect to time by performing a random walk on the interpolation rate. The interpolation is $1 + \varepsilon$ input sample per output sample, and ε is set near zero.
- Center frequency offset model: the offset Hz performing a random walk process is added to the incoming signal by a mixer.
- Noise model: simulates AWGN as well as frequency and timing offsets between the transmitter and receiver. The noise is added at the receiver side at a specific level according to the desired SNR.
- Fading model: uses the sum of sinusoids method for the number of expected multipath components. This block also takes in the Doppler frequency shift as a normalized value and a random seed to the noise generator to simulate Rician and Rayleigh fading processes.

2.1.4 Packaging Data

The output stream of each simulation is randomly segmented into vectors as the original dataset with a sample rate of 1M sample per second. The visualized time domain of samples for each modulation type is shown in Figure 2.2. We can easily tell the difference between an analog and a digital signal, but the difference among digital signals are not visually discernible. Similar to the way that an acoustic signal is windowed in voice recognition tasks, a slide window extracts 128 samples with a shift of 64 samples, which forms the new dataset we are using. A common form of input data in the machine learning community is $N_{samples} \times N_{channels} \times Dim_1 \times Dim_2$. The $N_{samples}$ in this study is 1200000 samples. $N_{channels}$ is usually three representing RGB for the image recognition task, but in a communication system, it is treated as one. Each sample consists of a 128 float32 array corresponding to the sample rate. Modulated signals are typically decomposed into in-phase and quadrature components, which can be a simple and flexible expression. Thus we have Dim_1 as 2 for the

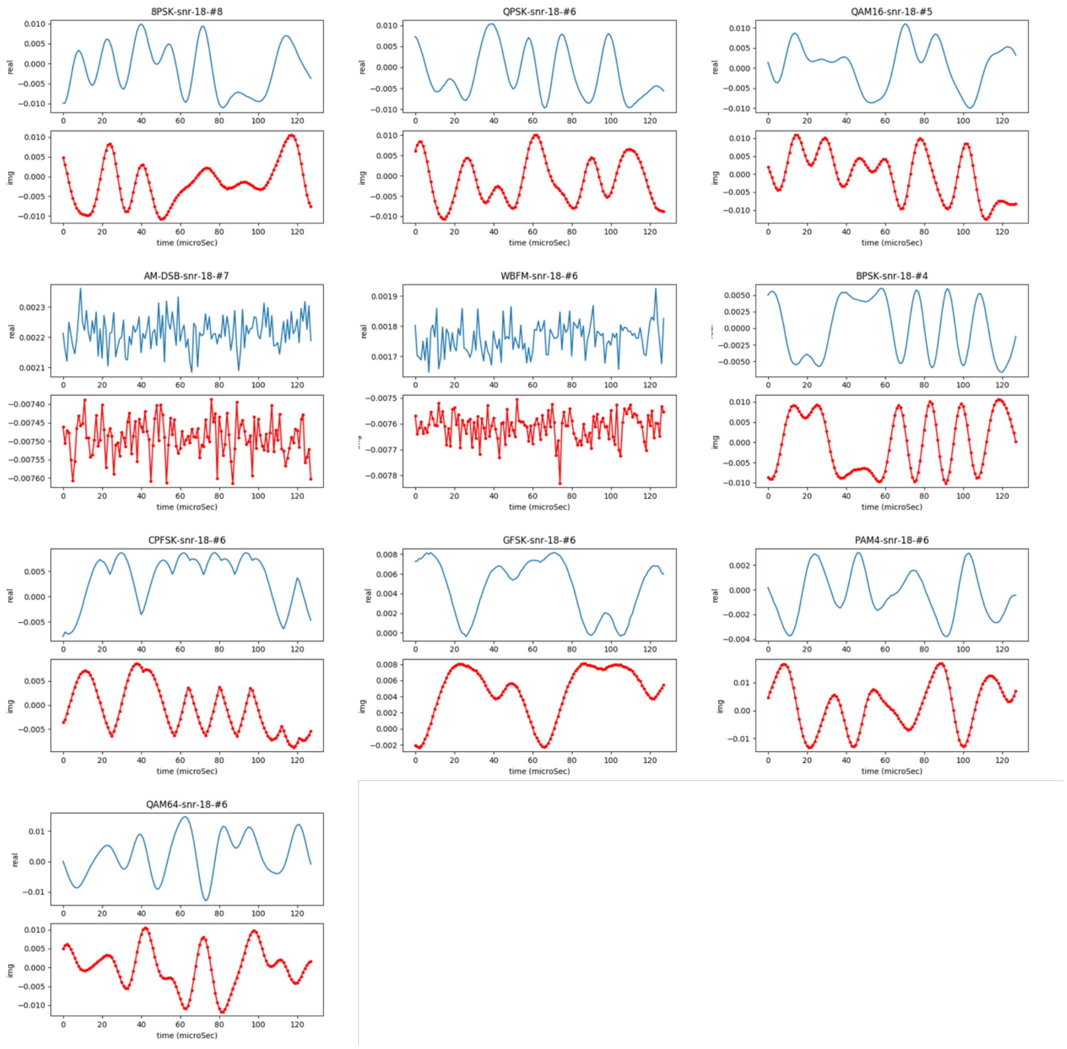


Fig. 2.2. Time domain visualization of the modulated signals

IQ components and Dim_2 as 128 holding the time dimension. The segmented samples represent the modulated schemes, the channel states and the random processes during signal propagation. As we focus on the task of modulation classification, we use the modulation schemes as labels for the samples. So the label input would be a 1×10 vector consisting of the 10 simulated modulation types.

2.2 Hardware

The training and testing experiments are conducted on two types of GPUs successively. Nvidia M60 GPU was firstly used for training basic neural networks and fine tuning. Later experiments were conducted on Tesla P100 GPU. All GPUs performance are maximized and the volatile GPU is fully utilized. The cuda and cudnn versions are 9.1.85 and 5.1.5, respectively. The framework of the preprocessing and the neural network codes are built using Keras with Theano and Tensorflow as backends.

3. NEURAL NETWORK ARCHITECTURE

The carrier frequency, phase offset and symbol synchronization are firstly recovered using moment based estimations or envelopes for all signals before the demodulation. Then convolution filters are applied for received signals to average out impulsive noises and optimize the SNR. Inspired by the fact that expert designed filters generally learn features from recovered signals, we use a convolutional neural network (CNN) to extract temporal features to form a robust feature basis. Various types of neural network architectures have been studied for image classification tasks, which are robust to the images rotation, occlusion, scaling and other noise conditions. Therefore, we applied several neural networks here to improve the blind modulation classification task which faces similar feature variations. We randomly choose half of the 1200000 examples for training and the other half for testing in each experiment.

The performance of a good classifier or a good neural network model is supposed to correctly decide the true modulation of an incoming signal from a pool of N_{mod} schemes. Let $P(i'|i)$ denotes the probability that the i^{th} modulation type is recognized as the i'^{th} one in the pool. For $i, i' = 1, ..., N_{mod}$, the probabilities can form a $N_{mod} \times N_{mod}$ confusion matrix, where the diagonal $P(i|i)$ represents the correctness of each modulation format. The average classification accuracy is then given by

$$P_c = N_{mod}^{-1} \textstyle\sum_{i=1}^{N_{mod}} P(i|i). \tag{3.1}$$

One can also use the complementary the expression of success rate to measure the performance, i.e. $P_e(i'|i) = 1 - P(i'|i)$. Here we use the previous one as the performance measure of our deep neural network architectures.

3.1 CNN

3.1.1 Architecture

CNNs are feed forward neural networks that pass the convolved information from the inputs to the outputs in only one direction. They are generally similar to the traditional neural networks and usually consist of convolutional layers and pooling layers as a module, but neurons in convolutional layers are connected to only part of the neurons in the previous layer. Modules stack on top of each other and form a deep network. Either one or two fully connected layers follow the convolutional modules for the final outputs. Based on the framework proposed in [1], we build a CNN model with similar architecture but different hyper-parameters (Figure 3.1). This network is also roughly similar to the one that works well on the MNIST image recognition task. In this pipeline, the raw vectors are input directly into a convolutional layer consisting of 256 filters that have the size of 1 × 3 each. Each filter convolves with 1 × 3 elements in the input vector and slides one step to the next 1 × 3 elements.

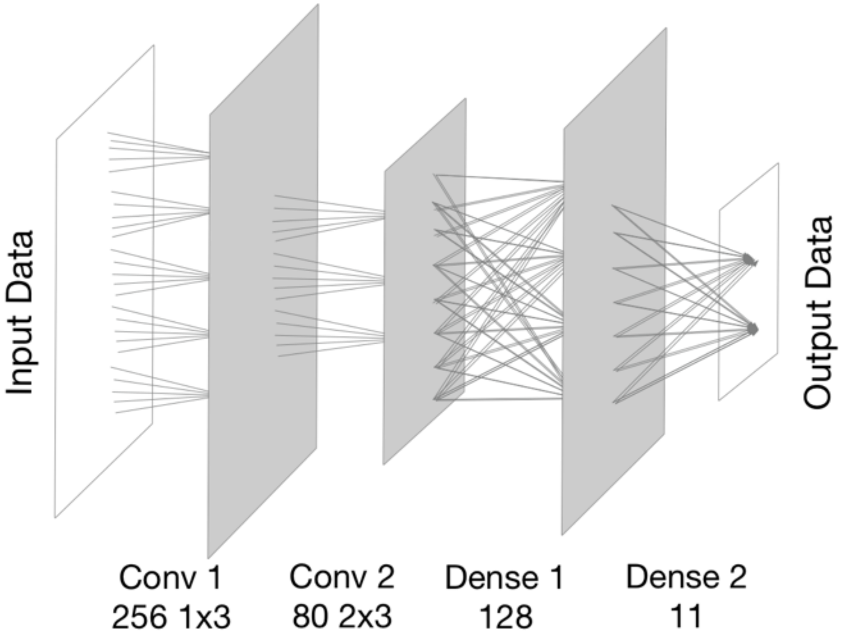


Fig. 3.1. Two-convolutional-layer model #1

Outputs of the first convolutional layer are then fed into the second convolutional layer that utilizes 80 filters with the size of 2×3. The outputs of the convolutional module are passed to the fully connected layers with 128 neurons and 11 neurons, with respect to order.

Although we send the signal segments in the form of 2×128 vectors that represent the in-phase and quadrature components of signals, the neural network regards the input as images with a resolution of 2×128 and only one channel. So filters in the convolutional module serve as feature extractors and learn the feature representations of the input 'images'. The neurons in convolutional layers are organized into the same number of feature maps as that of the filters. Each neuron in a filter is connected to a neighborhood of neurons in the previous layer through trainable weights, which is also call a receptive field or a filter bank [55]. Feature maps are generated from the convolution of the inputs with the learned weights, and the convolved results are sent through nonlinear functions (activation functions) for high dimensional mapping. Weights of all neurons within the same filters are constrained to be equal, whereas filters within the same convolutional layer have different weights. Therefore, multiple features can be extracted at the same location of an image through convolutional modules. A formal way for expressing this precess is

$$Y_k = f(W_k * x), \tag{3.2}$$

so the k^{th} feature map Y_k is derived from the 2D convolution of the related filter (W_k) and input (x), and $f(\cdot)$ is the nonlinear activation function.

In our model, all layers before the last one use rectified linear (ReLU) functions as the activation functions and the output layer uses the Softmax activation function to calculate the predicted label. ReLU was proposed by Nair and Hinton [56] in 2010 and popularized by Krizhevsky et al. [52]. The ReLU is given by $f(x) = max(x, 0)$, a simplified form of traditional activation functions such as sigmoid and hyperbolic tangent. The regularization technique to overcome overfitting includes normalization and dropout. Here, we set the dropout rate to 0.6 so that each hidden neuron in the network would be omitted at a rate of 0.6 during training. During the training phase, each epoch takes roughly 71s with the batch size of 1024. We do observe some overfitting as the validation loss inflects as the training loss decreases. We set the patience at 10 so that if the validation loss does not decline in 10 training epochs, the training would be regarded as converging and end. The total training time is roughly two hours for this model with Adam [57] as the optimizer.

The average classification accuracy for this model is 72% when the SNR is larger than 10dB. To further explore the relationship between the neural network architecture and success rate, we adjust the first model to a new one as illustrated in Figure 3.2. We exchange the first and second convolutional layer while keeping the remaining fully connected module the same as the first model. So the inputs pass through 80 large filters (size of 2 × 3) and 256 small filters (size of 1 × 3) subsequently. As the feature extraction process becomes sparse feature maps followed by relatively dense feature maps, the accuracy at high SNR increases to 75%. A natural hypothesis is that the convolutional layer with large, sparse filters extracting course grained fea-

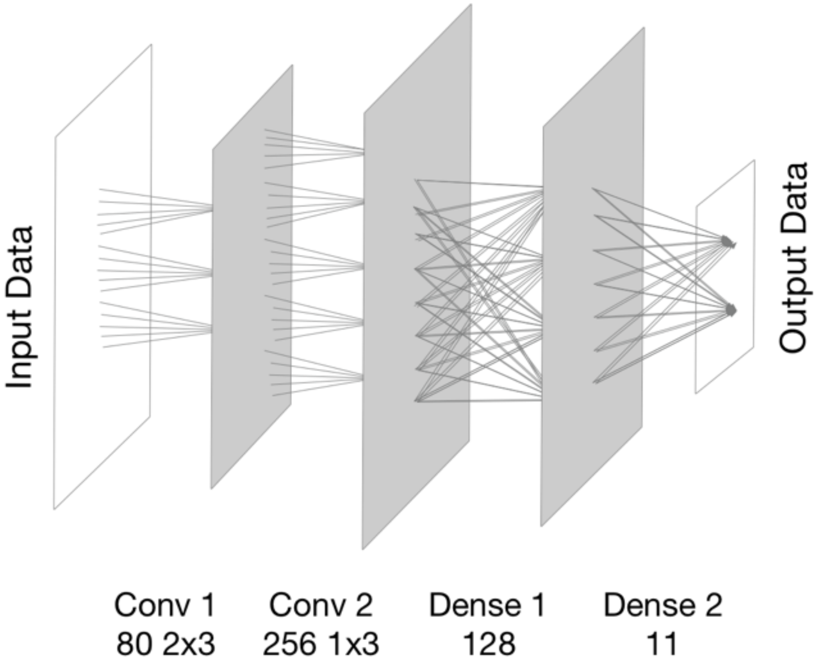


Fig. 3.2. Two-convolutional-layer model #2

tures followed by convolutional layers extracting fine grained features would produce better results. The training results of these models would be further discussed in the next subsection.

As shown by previous research for image recognition-related applications, deep convolutional neural networks inspired by CNNs have been one of the major contributors to architectures that enjoy high classification accuracies. The winner of the ILSVRC 2015 used an ultra-deep neural network that consists of 152 layers [58]. Multiple stacked layers were widely used to extract complex and invisible features, so we also tried out deeper CNNs that have three to five convolutional layers with two fully connected layers. We build a five-layer CNN model based on the one in Figure 3.2, but add another convolutional layer with 256 1 × 3 filters in front of the convolutional module. The average accuracy at high SNR is improved by 2%. The best classification accuracy is derived from the six-layer CNN as illustrated in Figure 3.3, where the layer with more and the largest filters is positioned at the second layer. The seven-layer CNN that performs best is produced by the architecture in Figure 3.4. As the neural network becomes deeper, it also gets harder for the validation loss to decrease. Most eight-layer CNNs see the validation loss diverge, and the only one

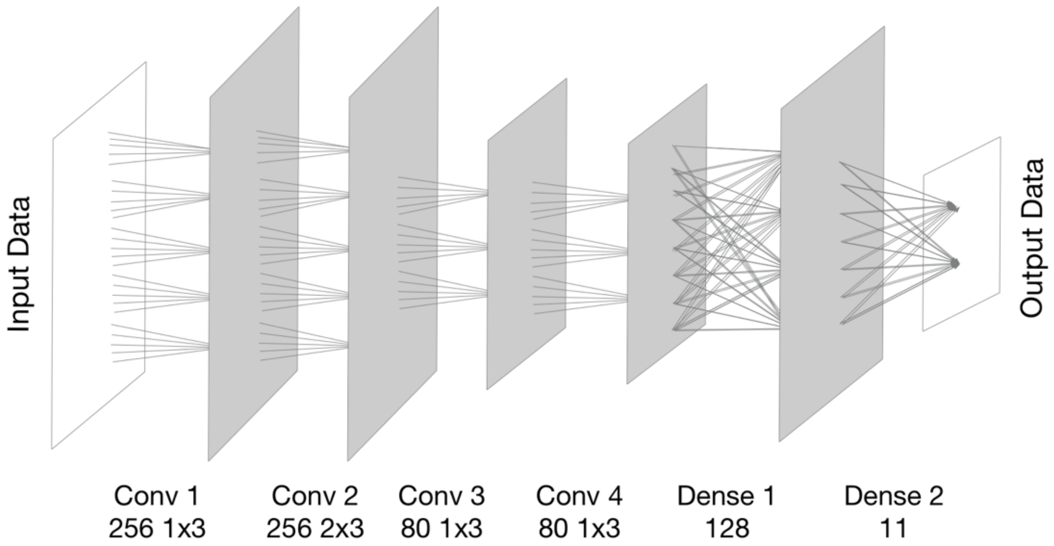


Fig. 3.3. Four-convolutional-layer model

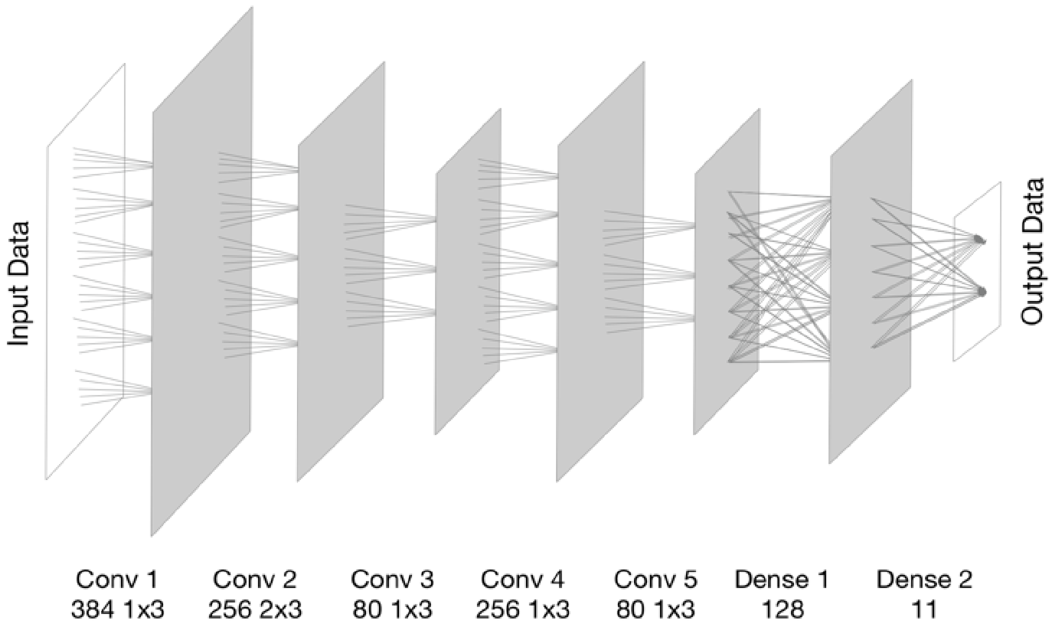


Fig. 3.4. Five-convolutional-layer model

that converges performs worse than the seven-layer CNN. The training time rises as the model becomes more complex, from 89s per epoch to 144s per epoch.

3.1.2 Results

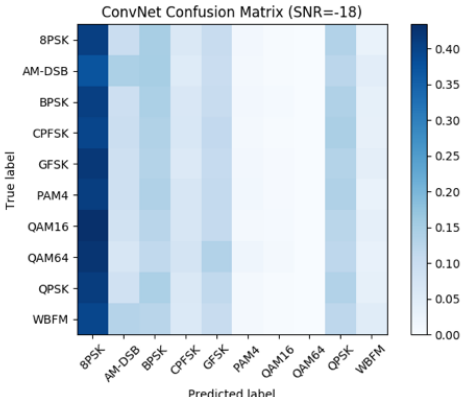


Fig. 3.5. Confusion matrix at -18dB SNR

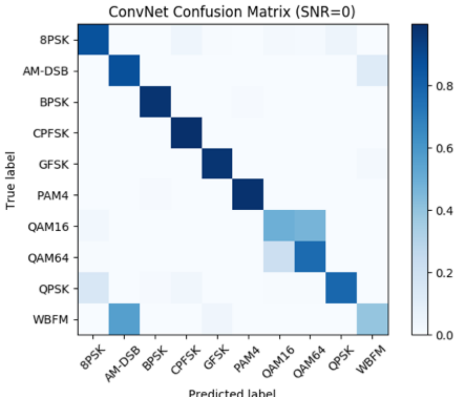


Fig. 3.6. Confusion matrix at 0dB SNR

We use 600000 samples for training and 600000 samples for testing. The classification results of our first model, four-layer CNN, is shown in forms of confusion matrix. In situations that signal power is below noise power, as for the case when the SNR is -18dB (Figure 3.5), it is hard for all neural networks to extract the desired signal features, while when SNR grows higher to 0dB, there is a prominent diagonal in the confusion matrix, denoting that most modulations are correctly recognized. As mentioned above, the highest average classification accuracy is produced by the CNN with four convolutional layers. In its confusion matrix (Figure 3.7), there is a clean diagonal and several dark blocks representing the discrepancies between WBFM and AM-DSB, QAM16 and QAM64, and 8PSK and QPSK. The training and testing data sets contain samples that are evenly distributed from -20dB SNR to +18 dB SNR. So we plot the prediction accuracy as a function of SNRs for all our CNN models. When the SNR is lower than -6dB, all models perform similar and it is hard to distinguish

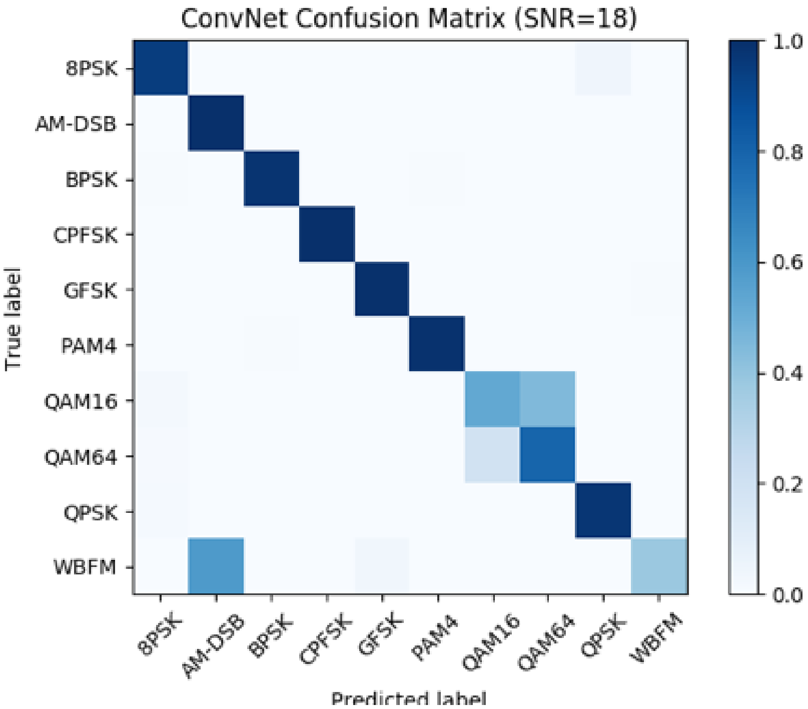


Fig. 3.7. Confusion matrix of the six-layer model at +16dB SNR

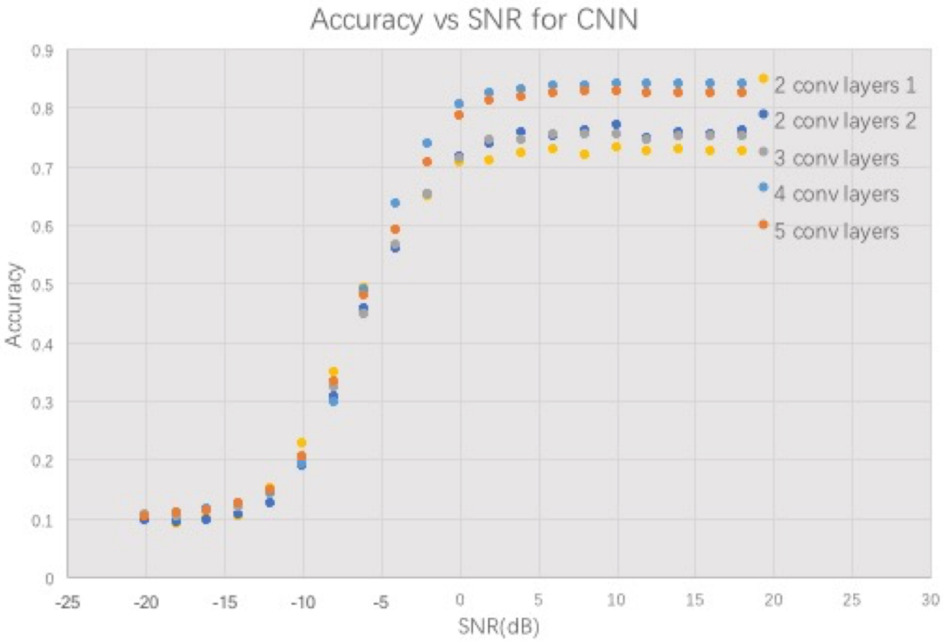


Fig. 3.8. Classification performance vs SNR

the modulation formats, while as the SNR becomes positive, there is a significant difference between deeper models and the original ones. The deepest CNN which utilizes five convolutional layers achieves 81% at high SNRs which is slightly lower than the 83.8% produced by the four-convolutional-layer model.

3.1.3 Discussion

Blank inputs or inputs that are exactly the same but with different labels can confuse neural networks since the neural network adjusts weights to classify it into one label. The misclassification between two analogue modulations is caused by the silence in the original data source. All samples with only the carrier tone are labeled as AM-DSB during training, so silence samples in WBFM are misclassified as AM-DSB when testing. In the case of digital signal discrepancies, different PSK and different QAM modulation types preserve similar constellation maps so it is difficult for CNNs to find the different features.

For neural networks deeper than eight layers, the large gradients passing through the neurons during training may lead to having the gradient irreversibly perish. The saturated and decreasing accuracy as the depth of the CNN grows is a commonly faced problem in deep neural network studies. However, there should exist a deeper model when it is constructed by copying the learned shallower model and adding identity mapping layers. So we explored a new architecture as discussed below.

3.2 ResNet

3.2.1 Architecture

Deep residual networks [59] led the first place entries in all five main tracks of the ImageNet [58] and COCO 2015 [60] competitions. As we see in the previous deep CNN training, the accuracy saturates or decreases rapidly when the depth of a CNN grows. The residual network solves this by letting layers fit a residual mapping. A building block of a residual learning network can be expressed as the function in Figure 3.9, where x and $H(x)$ are the input and output of this block, respectively. Instead of finding the mapping function $H(x) = x$ which is difficult in a deep network,

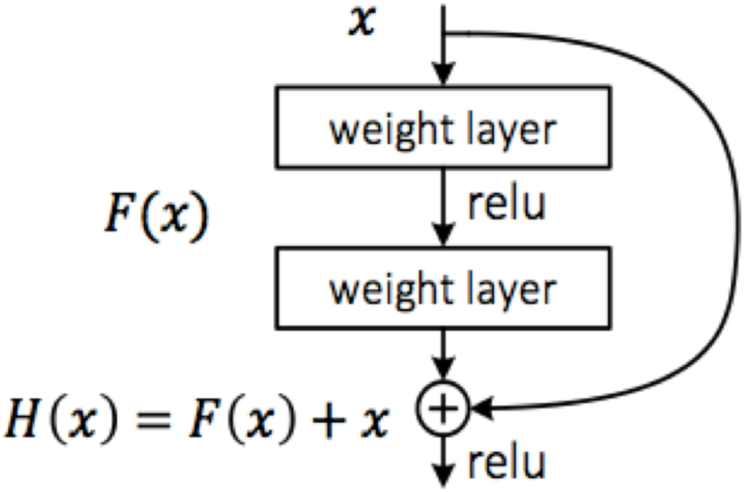


Fig. 3.9. A building block of ResNet

the ResNet adds a shortcut path so that it now learns the residual mapping function $F(x) = H(x) - x$. $F(x)$ is more sensitive to the input than $H(x)$ so the training of deeper networks becomes easy. The bypass connections create identity mappings so that deep networks can have the same learning ability as shallower networks do. Our neural network using the residual block is shown in Figure 3.10. It is built based on the six-layer CNN that performs best. Limited by the number of convolutional layers in the CNN model, we add only one path that connects the input layer and the third layer. The involved network parameters are increased due to the shortcut path so the training time grows to 980s per epoch.

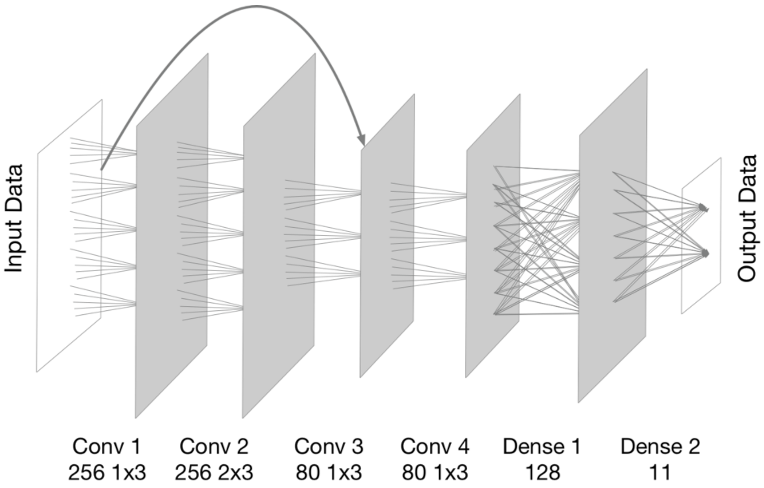


Fig. 3.10. Architecture of six-layer ResNet

3.2.2 Results

The classification accuracy as a function of SNR of the ResNet model displays the same trend as the CNN models. At high SNR, the best accuracy is 83.5% which is also similar to the six-layer CNN. However, when the depth of the ResNet grows to 11 layers, the validation loss does not diverge as the CNN model does, but produces a best accuracy of 81%.

3.2.3 Discussion

ResNet experiments on image recognition point out that the advantages of ResNet is prominent for very deep neural networks such as networks that are deeper than 50 layers. So it is reasonable that ResNet performs basically the same as CNNs when there are only six or seven layers. But it does solve the divergence problem in CNNs by the shortcut path. We tried another architecture that also uses bypass paths between different layers.

3.3 DenseNet

3.3.1 Architecture

The densely connected network (DenseNet) uses shortcut paths to improve the information flow between layers but in a different way from the ResNet. DenseNet solves the information blocking problem by adding connections between a layer and all previous layers. Figure 3.11 illustrates the layout of DenseNet for the three channel image recognition, where the l^{th} layer receives feature maps from all previous layers, x_0,...,x_{l-1} as input:

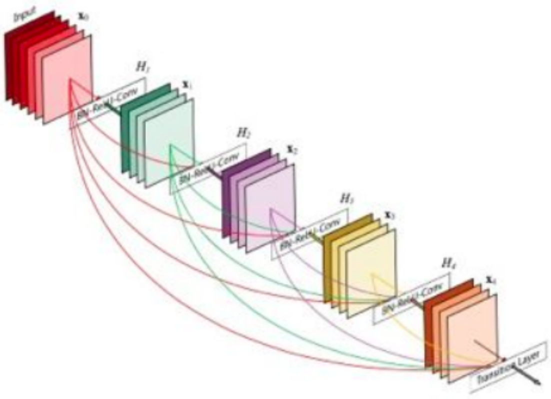


Fig. 3.11. The framework of DenseNet

$$x_l = H_l([x_0, x_1, ..., x_{l-1}]), \tag{3.3}$$

where H_l is a composite function of batch normalization, ReLU and Conv.

We implement the DenseNet architecture into our CNNs with different depths. Since there should be at least three convolutional layers in the densely connected module, we start from the three-convolutional-layer CNN (Figure 3.12). In this model, we add a connection between the first layer and the second one so that the output of the first convolutional layer is combined with the convolution results after the second layer and sent to the third layer. There is only one shortcut path in the model, which is also the case of ResNet, but the connections are created between different layers.

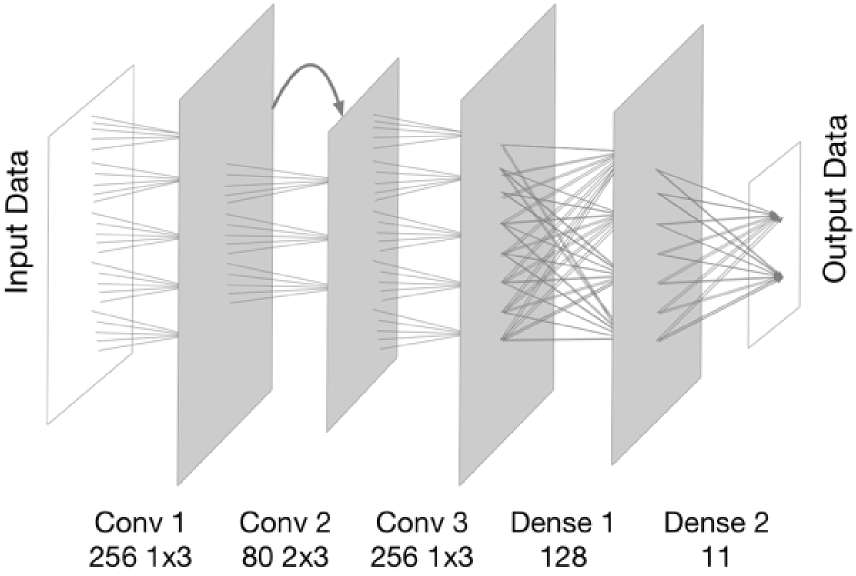


Fig. 3.12. Five-layer DenseNet architecture

Six-layer and seven-layer DenseNets are illustrated in Figure 3.13 and Figure 3.14, respectively. A DenseNet block is created between the first three convolutional layers in the six-layer CNN. The feature maps from the first and second layers are reused in the third layer. The training time remains relatively high at 1198s per epoch for all DenseNets.

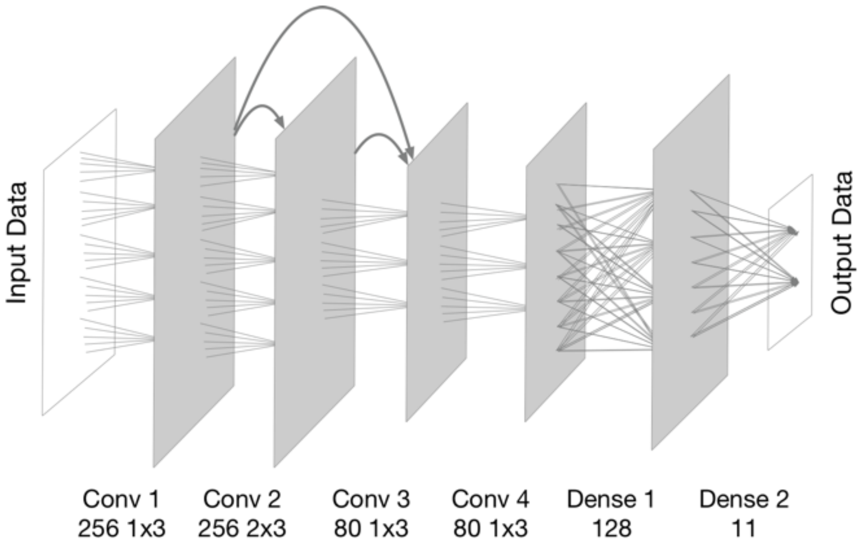


Fig. 3.13. Six-layer DenseNet architecture

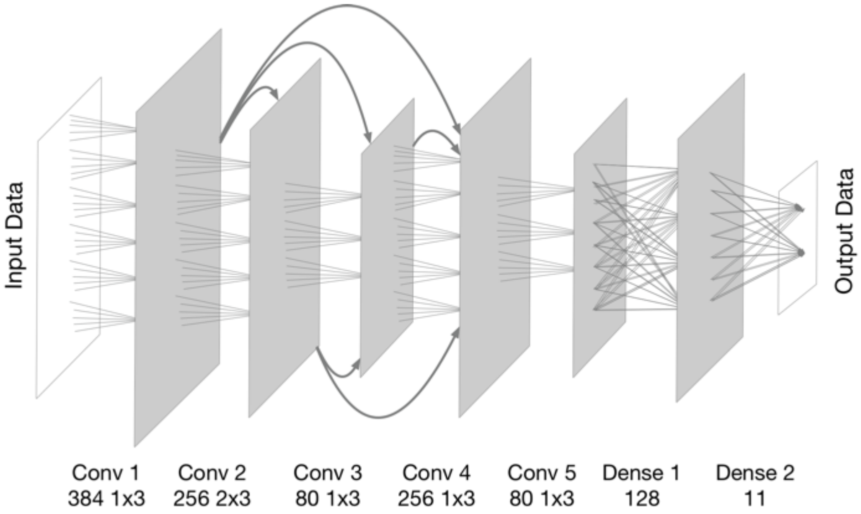


Fig. 3.14. Seven-layer DenseNet architecture

3.3.2 Results

The average accuracy of the seven-layer DenseNet improves 3% compared with that of the seven-layer CNN, while both accuracies vary in the same trend as functions of SNRs. In Figure 3.15, the DenseNet with four convolutional layers outperforms others with the accuracy at 86.8% at high SNRs, which is 3% higher than the accuracy of four-convolutional layer CNN. However, the average accuracy saturates when the depth of DenseNet reaches six.

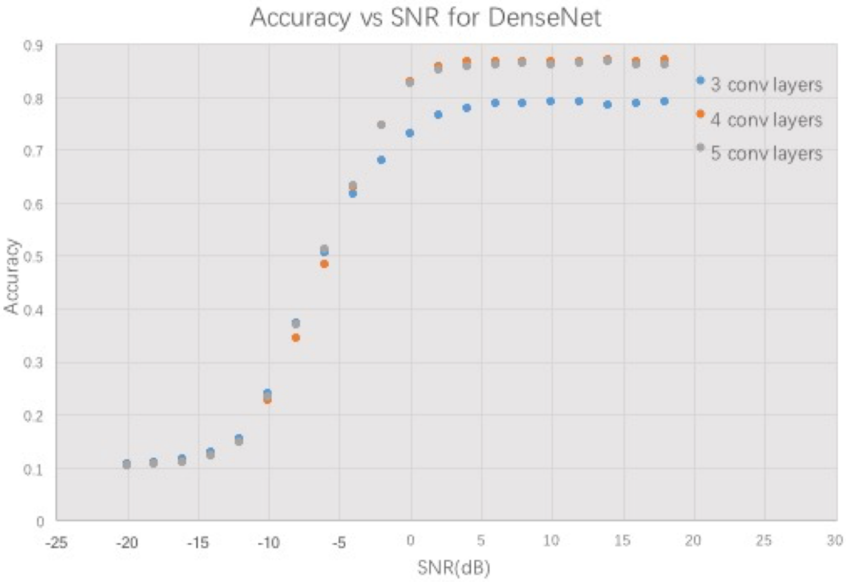


Fig. 3.15. Best Performance at high SNR is achieved with a four convolutional-layer DenseNet

3.3.3 Discussion

Although the ResNet and DenseNet architectures also suffer from accuracy degradation when the network grows deeper than the optimal depth, our experiments still show that when using the same network depth, DenseNet and ResNet have much higher convergence rates than plain CNN architectures. Figure 3.16 shows the validation errors of ResNet, DenseNet, and CNN of the same network depth with respect to the number of training epochs used. We can see that the ResNet and the DenseNet start at significantly lower validation errors and remain having a lower validation error throughout the whole training process, meaning that combining ResNet and DenseNet into a plain CNN architecture does make neural networks more efficient to train for the considered modulation classifcation task.

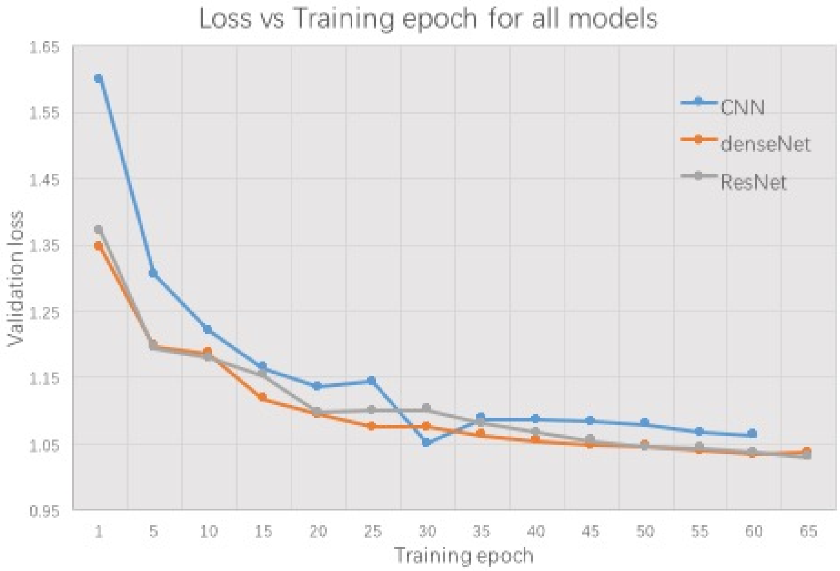


Fig. 3.16. Validation loss descents quickly in all three models, but losses of DenseNet and ResNet reach plateau earlier than that of CNN

3.4 CLDNN

3.4.1 Architecture

The Convolutional Long Short-Term Memory Deep Neural Network (CLDNN) was proposed by Sainath et al. [61] as an end-to-end model for acoustic learning. It is composed of sequentially connected CNN, LSTM and fully connected neural networks. The time-domain raw voice waveform is passed into a CNN, then modeled through LSTM and finally resulted in a 3% improvement in accuracy. We built a similar CLDNN model with the architecture in Figure 3.17, where a LSTM module with 50 neurons is added into the four-convolutional CNN. This architecture that captures both spacial and temporal features is proved to have superior performance than all previously tested architectures.

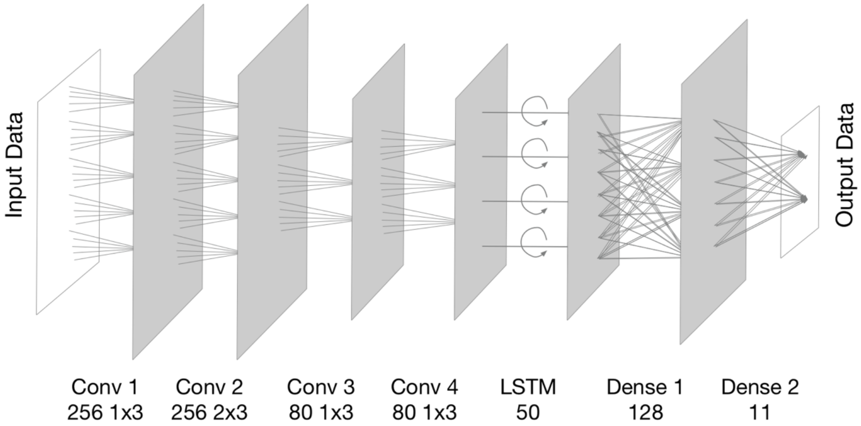


Fig. 3.17. Architecture of the CLDNN model

3.4.2 Results

The best average accuracy is achieved by the CLDNN model at 88.5%. In Figure 3.18, we can see that CLDNN outperforms others across almost all SNRs. The cyclic connections in LSTM extract features that are not obtainable in other architectures.

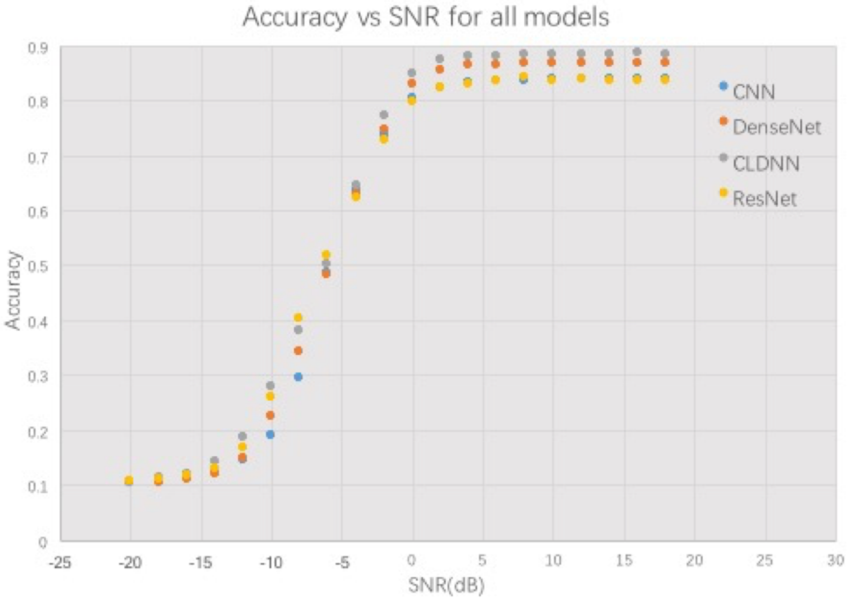


Fig. 3.18. Classification performance comparison between candidate architectures.

3.4.3 Discussion

The CNN module in CLDNN extracts spacial features of the inputs and the LSTM module captures the temporal characters. CLDNN has been highly accepted in speech recognition tasks, as the CNN, LSTM and DNN modules being complementary in the modeling abilities. CNNs are good at extracting location information, LSTMs excels at temporal modeling and DNNs are suitable for mapping features into a separable space. The combination was first explored in [62], however the CNN, LSTM and DNN are trained separately and the three output results were combined through a combination layer. In our model, they are unified in a framework and trained jointly. The LSTM is inserted between CNN and DNN because it is discovered to perform better if provided higher quality features. The characteristic causality existing in modulated signals that is the same as the sequential relationship in natural languages contributes the major improvements of the accuracy.

Table 3.1.
Significant modulation type misclassification at high SNR for the proposed CLDNN architecture

Misclassification	**Percentage(%)**
8PSK/QPSK	5.5
QAM64/QAM16	20.14
WBFM/AM-DSB	59.6
WBFM/GFSK	3.3

Although there is a significant accuracy improvement for all modulation schemes in the confusion matrix of the CLDNN model, there are still few significant confusion blocks existing off the diagonal. The quantified measures for these discrepancies are formed in Table 3.1. The confusion between WBFM and AM-DSB has the more prominent influence on the misclassification rate, but this is caused by the original data source and we cannot reduce it by simply adjusting neural networks. So we focus on improving the classification of QAM signals.

3.5 Cumulant Based Feature

3.5.1 Model and FB Method

As mentioned above, an intuitive solution for the misclassification is to separate the classification of QAM from the main framework. So a new model is proposed in the thesis, which labels both QAM16 and QAM64 as QAM16 during training. At the testing phase, if the input example is decided as QAM16, it would be sent to another classifier which classifies QAM16 and QAM64. A pipeline is illustrated in Figure 3.19 below We still use the trained CLDNN as a main framework, and explore feature based methods as M2 which will be introduced below for QAM16 and QAM64, since the tested neural networks perform poor at QAM recognitions.

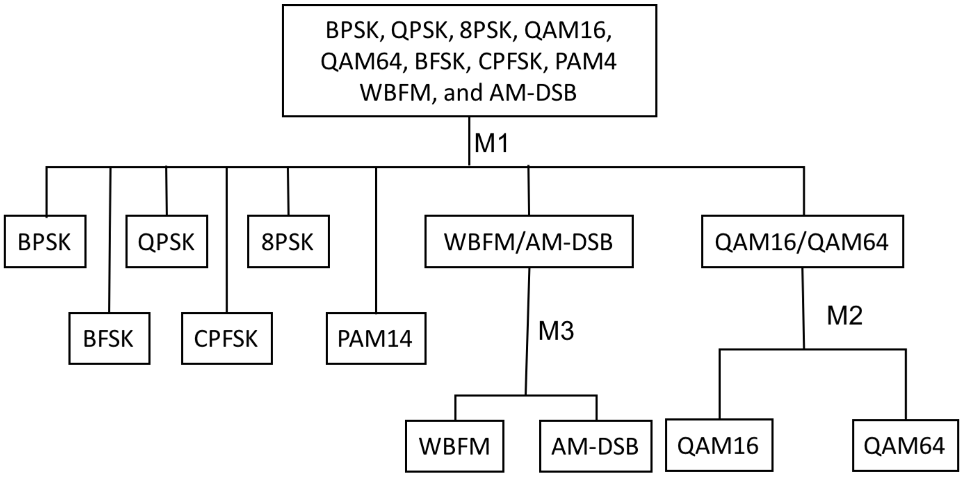


Fig. 3.19. Block diagram of the proposed method showing two stages

The first stage of the M2 is based on the pattern recognition approach, so cumulants are derived from QAM16 and QAM64 modulated signals as key features. Cumulants are made up of moments which are defined as

$$M_{pq} = E\left[y(k)^{p-q}y^*(k)^q\right], \tag{3.4}$$

where * is the conjugation. The cumulants for complex valued, stationary signal can be derived from moments. High order cumulants that are higher than the second order have the following advantages:

- The high order cumulant is always zero for colored Gaussian noise, namely it is less effected by the Gaussian background noises, so it can be used to extract non-Gaussian signals in the colored Gaussian noise,
- The high order cumulant contains system phase information, so it can be utilized for non-minimized phase recognition.
- It can detect the nonlinear signal characters or recognize nonlinear systems.

High order cumulants for stationary signals are defined as

$$C_{40} = cum(y(n), y(n), y(n), y(n)) = M_{40} - 3M_{20}^2, \tag{3.5}$$

$$C_{41} = cum(y(n), y(n), y(n), y^*(n)) = M_{41} - 3M_{20}M_{21}, \tag{3.6}$$

$$C_{42} = cum(y(n), y(n), y^*(n), y^*(n)), \tag{3.7}$$

$$\begin{aligned} C_{61} &= cum(y(n), y(n), y(n), y(n), y(n), y^*(n)) \\ &= M_{61} - 5M_{21}M_{40} - 10M_{20}M_{41} + 30M_{20}^2M_{21}, \end{aligned} \tag{3.8}$$

$$\begin{aligned} C_{62} &= cum(y(n), y(n), y(n), y(n), y^*(n), y^*(n)) \\ &= M_{62} - 6M_{20}M_{42} - 8M_{21}M_{41} - M_{22}M_{40} + 6M_{20}^2M_{22} + 24M_{21}^2M_{20}. \end{aligned} \tag{3.9}$$

Given the cumulants of QAM16 and QAM64 modulated signals, a SVM is applied as a binary classifier with a RBF as the kernel function. The input of the SVM is a set of features containing the signal information. Here we use the cumulants, SNR and time indexes forming 1 × 3 vectors in the second stage of M2.

3.5.2 Results and Discussion

We use the high order cumulants C_{63} as feature statistics. In Figure 3.20, each fifty samples are averaged and cumulated to produce a C_{63}, so Figure 3.20 depicts the cumulants of QAM16 and QAM64 modulated signals as functions of time. There are obvious distinctions between QAM16 and QAM64 modulated signals over a short period, but both cumulants fluctuate across time. Previous studies use cumulants as the key features based on the assumption that the signals are stationary, so the cumulants remain stable during a long time period. However, that is not the case in our study as the simulated signals are not stationary. So we add the time index as one of the key features for the SVM to learn, since it is discernible that the cumulants are constant during a short period of time. SNRs are also utilized as one of the key

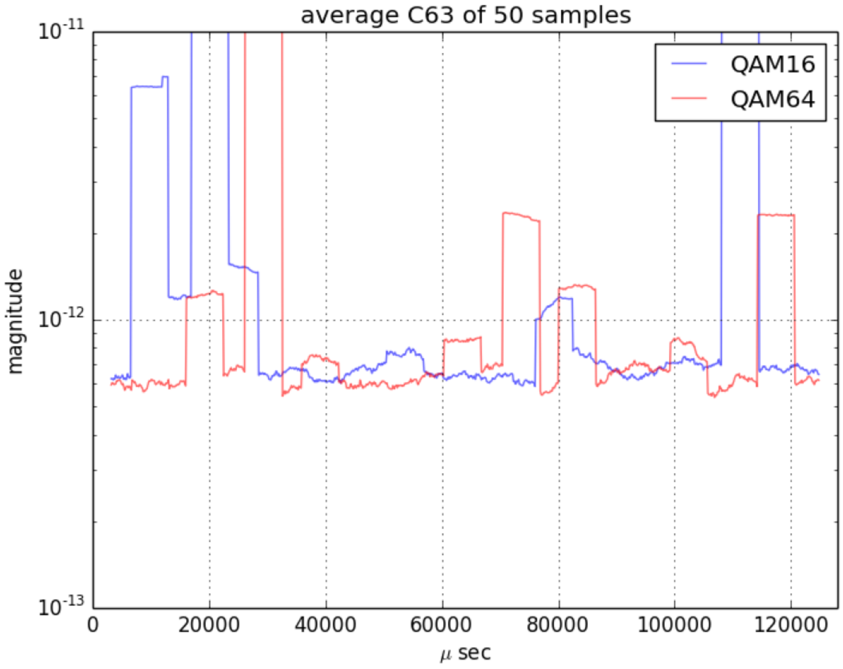


Fig. 3.20. The cumulants of QAM16 and QAM64 modulated signals with respect to time

features because models developed for a specific SNR are not adaptable for other SNRs.

The best binary classification accuracy was obtained using the default penalty parameter C and gamma in the RBF kernel, which is 27%. For a binary classifier, the classification accuracy ranges from 50% to 100%. By flipping the labels during training, we get a 72% classification roughly across all SNRs. With the same recognition rates of modulations in CLDNN but higher QAM success rate, the average classification accuracy of this model reaches roughly 90%.

3.6 LSTM

3.6.1 Architecture

Recurrent neural networks (RNN) are commonly regarded as the starting point for sequence modeling [63] and widely used in translation and image captioning tasks. The most significant character in them is allowing information to persist. Given an input vector $\boldsymbol{x} = (x_1, ..., x_T)$, the hidden layer sequence $\boldsymbol{h} = (h_1, ..., h_T)$ and the output sequence $\boldsymbol{y} = (y_1, ..., y_T)$ in a RNN can be iteratively computed using

$$h_t = \ H \ (W_{ih}x_t \ + \ W_{hh}h_t - 1 \ + \ b_h), \tag{3.10}$$

$$y_t \ = \ W_{h0}h_t \ + \ b_0, \tag{3.11}$$

where W denotes the weight matrix between the i^{th} and the h^{th} layers, b is the bias vector and H is the activation function in hidden layers. In RNN, the outputs of last time steps are reused as the inputs of the new time step, which connects previous information to the present task, such as using previous words might inform the understanding of the present word. However, the memory period cannot be controlled in RNNs and they also have the gradient decent problem. The LSTM are designed to avoid the long-term problem [64]. The LSTM does have the ability to remove or add information to the neuron state, carefully regulated by structures called gates. Gates are designed like the memory units that can control the storage of previous outputs. Figure 3.21 shows the architecture of a LSTM memory cell, which is composed of three gates: input gate, forget gate and output gate. The H activation function used in this cell is implemented by a composite function:

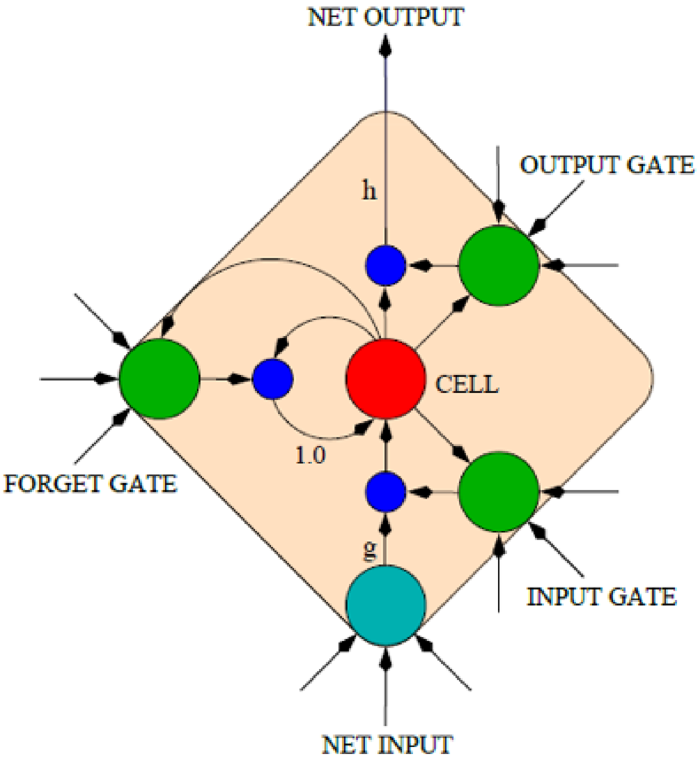


Fig. 3.21. Architecture of the memory cell

$$i_t = \sigma\,(W_{xi}xt_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), \tag{3.12}$$

$$f_t = \sigma\,(W_{xf}xt_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f), \tag{3.13}$$

$$o_t = \sigma\,(W_{xo}t_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o), \tag{3.14}$$

$$c_t = f_tc_{t-1} + i_t \tan h\,(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \tag{3.15}$$

$$h_t = o_t\ \tan h(c_t), \tag{3.16}$$

where i, f, o, c are the input gate, forget gate, output gate and cell activation vectors, respectively, σ is the logistic activation function, and W is the weight matrix with the subscript representing the corresponding layers. The forget gate controls the length of memory and the output gate decides the output sequence.

Previous study [65] has conducted LSTM classifier experiments which used a smaller dataset and got the accuracy of 90%. Our study uses a larger dataset and fine tune the model by adjusting the hyperparameters to produce better results. The LSTM architecture is described in a flow chart in Figure 3.22. Here we preprocessed the input data which originally use IQ coordinates, where the in-phase and quadrature components are expressed as $I = A\cos(\phi)$ and $Q = A\sin(\phi)$. We format the IQ into time domain A and ϕ and pass the instantaneous amplitude and phase information of the received signal into the LSTM model. Samples from $t - n$ to t are sent sequentially and the two LSTM layers extract the temporal features in amplitude and phase, followed by two fully connected layers.

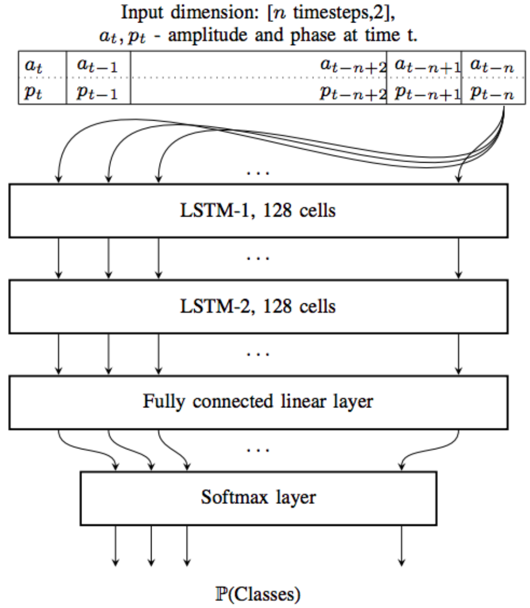
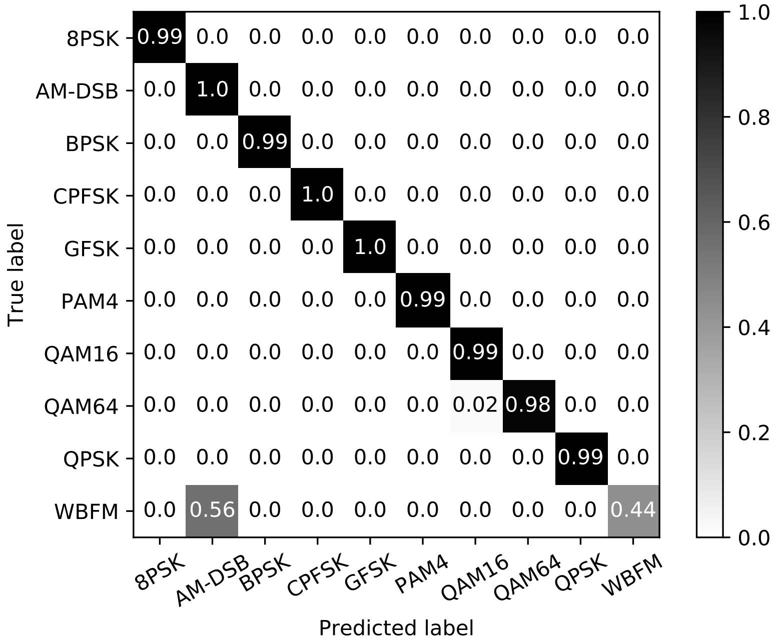


Fig. 3.22. Architecture of CLDNN model

3.6.2 Results and Discussion

The classification accuracies across all modulations are presented in the confusion matrix (Figure 3.23). All modulations except WBFM are correctly recognized at a high accuracy, even the QAM 16 and QAM64, BPSK and QPSK confusions are removed. The average accuracy reaches approximately 94%. Roughly half of WBFM samples are labeled as AM-DSB during testing, due to the silence in the source audio. We also input IQ samples directly into LSTM, which yield poor performance while the amplitude and phase inputs produced good results. The QAM16 and QAM64 classification accuracies failed to 1 and 0 when time domain IQ samples are fed into the LSTM model, as the training loss diverges during the training phase. An explanation for this result would be that LSTM layers are not sensitive to time domain IQ information. Suppose an instantaneous complex coordinates for the QAM16 signal is (x, y), and the coordinates for QAM64 is $(x + \triangle x, y)$. The difference in IQ format,



	8PSK	AM-DSB	BPSK	CPFSK	GFSK	PAM4	QAM16	QAM64	QPSK	WBFM
8PSK	0.99	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
AM-DSB	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BPSK	0.0	0.0	0.99	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CPFSK	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
GFSK	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
PAM4	0.0	0.0	0.0	0.0	0.0	0.99	0.0	0.0	0.0	0.0
QAM16	0.0	0.0	0.0	0.0	0.0	0.0	0.99	0.0	0.0	0.0
QAM64	0.0	0.0	0.0	0.0	0.0	0.0	0.02	0.98	0.0	0.0
QPSK	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.99	0.0
WBFM	0.0	0.56	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.44

Fig. 3.23. The confusion matrix of LSTM when SNR=+18dB

$\triangle x$, is too small to be captured by the network. While in amplitude and phase format, the difference in x direction can be decomposed into both amplitude and phase directions, which are easier to be observed by the neural network.

The performance comparison of all architectures that perform best across different depth is given in Figure 3.24. When SNR is less than -6dB, all architectures fail to perform as expected, but they all produce stable classification results at positive SNRs. Almost all of the highest accuracies are achieved when the SNR ranges from +14dB to +18dB.

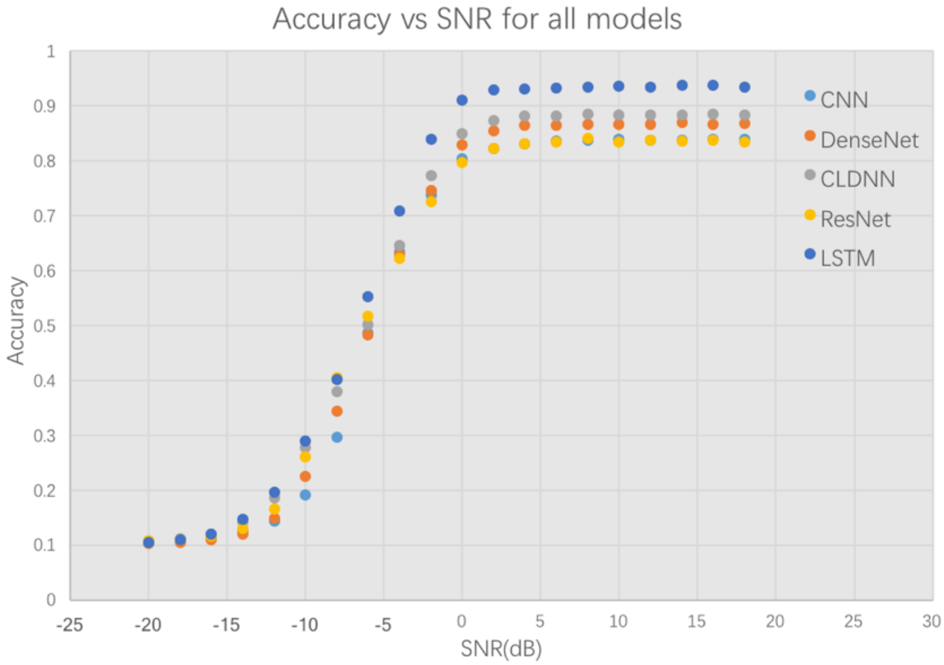


Fig. 3.24. Best Performance at high SNR is achieved by LSTM

4. CONCLUSION AND FUTURE WORK

4.1 Conclusion

This thesis have implemented several deep learning neural network architectures for the automatic modulation classification task. Multiple classifiers are built and tested, which provide high probabilities of correct modulation recognition in a short observation time, particularly for the large range of the SNR from -20dB to +18dB. The trained models outperform traditional classifiers by their high success rates and low computation complexities. The CNN serves as a basic end-to-end modulation recognition model providing nonlinear mapping and automatic feature extraction. The performance of CNNs are improved from 72% [1] to 83.3% by increasing the depth of CNNs. ResNet and DenseNet were used to build deeper neural networks and enhance the information flow inside the networks. The average classification accuracy reaches 83.5% and 86.6% for ResNet and DenseNet, respectively. Although the best accuracies are limited by the depth of network, they suggest that the shortcut paths between non-consecutive layers produce better classification accuracies. A CLDNN model combines a CNN block, a LSTM block and a DNN block as a classifier that can automatically extract the spacial and temporal key features of signals. This model produces the highest accuracy for time domain IQ inputs and can be considered as a strong candidate for dynamic spectrum access systems which highly relies on low SNR modulation classifications. The two-layer LSTM model was proposed with different time domain format inputs. The results reach roughly 100% for all digital modulations. The experiments of time domain IQ and amplitude phase inputs also emphasize the importance of preprocessing and input representation. These models are capable to recognizing the modulation formats with various propagation characteristic, and show high real-time functionality.

4.2 Future Work

An important consideration of the neural network performance measurement is the training time. Currently we run training on one high-performance GPU per experiment. The run time increases as the model becomes more complex. The total run time for a DenseNet or CLDNN could be three or four days. So further experiments should be deployed on multiple GPUs to shorten training time and take the best utilization of the GPU resources.

We have reached a high average classification accuracy of 94% with most modulations correctly recognized at high success rates. The misclassification between WBFM and AM-DSB could not be resolved by the classifier due to the source data. To address this problem, other modulated signals could be applied on our models to test the adaptive ability. Experimentally generated modulated signals from our lab would be applied on those models.

As we found in DenseNet and ResNet, the accuracy was limited by the depth of the networks and the deeper network cannot be trained as the training loss diverges. Further study could focus on this problem by adjusting hyper-parameters or preprocessing techniques.

REFERENCES

REFERENCES

[1] T. J. O'Shea and J. Corgan, "Convolutional radio modulation recognition networks," *CoRR*, vol. abs/1602.04105, 2016.

[2] K. Kim and A. Polydoros, "Digital modulation classification: the bpsk versus qpsk case," *Proc. IEEE Military Communications Conference (MILCOM)*, pp. 431–436, 1988.

[3] P. Sapiano and J. Martin, "Maximum likelihood psk classifier," *Proc. IEEE MILCOM*, vol. 3, pp. 1010–1014, 1996.

[4] L. Hong and K. Ho, "Classification of bpsk and qpsk signals with unknown signal level using the bayes technique," *Proc. IEEE International Symposium on Circuits and Systems*, vol. 4, pp. IV–IV, 2003.

[5] P. Panagiotou, A. Anastasopoulos, and A. Polydoros, "Likelihood ratio tests for modulation classification," *Proc. IEEE MILCOM*, vol. 2, pp. 670–674, 2000.

[6] N. E. Lay and A. Polydoros, "Per-survivor processing for channel acquisition, data detection and modulation classification," *IEEE Asilomar Conference on Signals, Systems, and Computers*, vol. 2, pp. 1169–1173, 1994.

[7] H. L. Van Trees, *Detection, estimation, and modulation theory, part I: detection, estimation, and linear modulation theory.* John Wiley & Sons, 2004.

[8] K. M. Chugg, C.-S. Long, and A. Polydoros, "Combined likelihood power estimation and multiple hypothesis modulation classification," *IEEE Asilomar Conference on Signals, Systems, and Computers*, vol. 2, pp. 1137–1141, 1995.

[9] N. E. Lay and A. Polydoros, "Modulation classification of signals in unknown isi environments," *Proc. IEEE MILCOM*, vol. 1, pp. 170–174, 1995.

[10] L. Hong and K. Ho, "Antenna array likelihood modulation classifier for bpsk and qpsk signals," *Proc. IEEE MILCOM*, vol. 1, pp. 647–651, 2002.

[11] O. Dobre, J. Zarzoso, Y. Bar-Ness, and W. Su, "On the classification of linearly modulated signals in fading channel," *Proc. Conference on Information Systems and Sciences*, vol. 26, 2004.

[12] W. Wei and J. M. Mendel, "Maximum-likelihood classification for digital amplitude-phase modulations," *IEEE Transactions on Communications*, vol. 48, no. 2, pp. 189–193, 2000.

[13] A. Polydoros and K. Kim, "On the detection and classification of quadrature digital modulations in broad-band noise," *IEEE Transactions on Communications*, vol. 38, no. 8, pp. 1199–1211, 1990.

[14] J. Sills, “Maximum-likelihood modulation classification for psk/qam,” *Proc. IEEE MILCOM*, vol. 1, pp. 217–220, 1999.

[15] C. Long, K. Chugg, and A. Polydoros, “Further results in likelihood classification of qam signals,” *Proc. IEEE MILCOM*, pp. 57–61, 1994.

[16] O. A. Dobre and F. Hameed, “Likelihood-based algorithms for linear digital modulation classification in fading channels,” *IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 1347–1350, 2006.

[17] F. Hameed, O. A. Dobre, and D. C. Popescu, “On the likelihood-based approach to modulation classification,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, 2009.

[18] A. I. Fontes, L. A. Pasa, V. A. de Sousa Jr, F. M. Abinader Jr, J. A. Costa, and L. F. Silveira, “Automatic modulation classification using information theoretic similarity measures,” *IEEE Vehicular Technology Conference (VTC), 9th*, pp. 1–5, 2012.

[19] A. K. Nandi and E. E. Azzouz, “Algorithms for automatic modulation recognition of communication signals,” *IEEE Transactions on Communications*, vol. 46, no. 4, pp. 431–436, 1998.

[20] X. Zhinan and B. Wenle, “A novel blind recognition algorithm for modulated m-qam signals,” *IEEE WRI International Conference on Communications and Mobile Computing*, vol. 1, pp. 461–465, 2009.

[21] M. P. DeSimio and G. E. Prescott, “Adaptive generation of decision functions for classification of digitally modulated signals,” *Proc. IEEE National Aerospace and Electronics Conference*, pp. 1010–1014, 1988.

[22] K. Ho, W. Prokopiw, and Y. Chan, “Modulation identification by the wavelet transform,” *Proc. IEEE MILCOM*, vol. 2, pp. 886–890, 1995.

[23] V. Ramakomar, D. Habibi, and A. Bouzerdoum, “Automatic recognition of digitally modulated communications signals,” *Proc. IEEE International Symposium on Processing and Its Applications*, vol. 2, pp. 753–756, 1999.

[24] J. J. Popoola and R. van Olst, “A novel modulation-sensing method,” *IEEE Vehicular Technology Magazine*, vol. 6, no. 3, pp. 60–69, 2011.

[25] W. Wei and J. M. Mendel, “A fuzzy logic method for modulation classification in nonideal environments,” *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 3, pp. 333–344, 1999.

[26] M. Pedzisz and A. Mansour, “Automatic modulation recognition of mpsk signals using constellation rotation and its 4th order cumulant,” *Digital Signal Processing*, vol. 15, no. 3, pp. 295–304, 2005.

[27] S. Gulati and R. Bhattacharjee, “Automatic blind recognition of noisy and faded digitally modulated mqam signals,” *IEEE India Conference*, pp. 1–6, 2006.

[28] C. Louis and P. Sehier, “Automatic modulation recognition with a hierarchical neural network,” *Proc. IEEE MILCOM*, pp. 713–717, 1994.

[29] H. Gang, L. Jiandong, and L. Donghua, "Study of modulation recognition based on hocs and svm," *IEEE VTC*, vol. 2, pp. 898–902, 2004.

[30] L. De Vito and S. Rapuano, "Validating a method for classifying digitally modulated signals," *Measurement*, vol. 42, no. 3, pp. 427–435, 2009.

[31] N. Ghani and R. Lamontagne, "Neural networks applied to the classification of spectral features for automatic modulation recognition," *Proc. IEEE MILCOM*, vol. 1, pp. 111–115, 1993.

[32] M. D. Wong and A. K. Nandi, "Automatic digital modulation recognition using spectral and statistical features with multi-layer perceptrons," *IEEE Signal Processing and its Applications*, vol. 2, pp. 390–393, 2001.

[33] S.-Z. Hsue and S. S. Soliman, "Automatic modulation classification using zero crossing," *IEEE Radar and Signal Processing*, vol. 137, no. 6, pp. 459–464, 1990.

[34] K. Ho, W. Prokopiw, and Chan, "Modulation identification of digital signals by the wavelet transform," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 147, no. 4, pp. 169–176, 2000.

[35] K. Maliatsos, S. Vassaki, and P. Constantinou, "Interclass and intraclass modulation recognition using the wavelet transform," *IEEE Personal, Indoor and Mobile Radio Communications*, pp. 1–5, 2007.

[36] C.-S. Park, J.-H. Choi, S.-P. Nah, W. Jang, and D. Y. Kim, "Automatic modulation recognition of digital signals using wavelet features and svm," *International Conference on Advanced Communications Technology*, vol. 1, pp. 387–390, 2008.

[37] L. De Vito, S. Rapuano, and M. Villanacci, "Prototype of an automatic digital modulation classifier embedded in a real-time spectrum analyzer," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 10, pp. 2639–2651, 2010.

[38] C. J. Le Martret and D. Boiteau, "Modulation classification by means of different orders statistical moments," *Proc. IEEE MILCOM*, vol. 3, pp. 1387–1391, 1997.

[39] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Transactions on Communications*, vol. 48, no. 3, pp. 416–429, 2000.

[40] M. Wong and A. Nandi, "Efficacies of selected blind modulation type detection methods for adaptive ofdm systems," *International Conferences on Signal Processing and Communication Systems*, 2007.

[41] B. Kosko, "Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence/book and disk," *Prentice hall*, 1992.

[42] L. Mingquan, X. Xianci, and L. Lemin, "Ar modeling-based features extraction of multiple signals for modulation recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 2, pp. 1385–1388, 1998.

[43] B. G. Mobasseri, "Digital modulation classification using constellation shape," *Signal processing*, vol. 80, no. 2, pp. 251–277, 2000.

[44] L. Mingquan, X. Xianci, and L. Leming, "Cyclic spectral features based modulation recognition," *Proc. IEEE International Conference on Communication Technology*, vol. 2, pp. 792–795, 1996.

[45] E. E. Azzouz and A. K. Nandi, "Modulation recognition using artificial neural networks," *Automatic Modulation Recognition of Communication Signals*, pp. 132–176, 1996.

[46] K. E. Nolan, L. Doyle, D. OMahony, and P. Mackenzie, "Modulation scheme recognition techniques for software radio on a general purpose processor platform," *Proc. Joint IEI/IEE Symposium on Telecommunication Systems, Dublin*, 2001.

[47] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[48] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances in Neural Information Processing Systems*, pp. 396–404, 1990.

[49] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[50] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2528–2535, 2010.

[51] K. Yu, Y. Lin, and J. Lafferty, "Learning image representations from the pixel level via hierarchical sparse coding," *CVPR*, pp. 1713–1720, 2011.

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[53] E. Blossom, "Gnu radio: Tools for exploring the radio frequency spectrum," *Linux J.*, no. 122, pp. 4–, Jun. 2004.

[54] T. OShea, "Gnu radio channel simulation," in *Proc. GNU Radio Conference*, 2013.

[55] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[56] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proc. International Conference on Machine Learning*, pp. 807–814, 2010.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR,vol. abs/1412.6980*, 2014.

[58] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CVPR*, pp. 770–778, 2016.

[60] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[61] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," *International Conference on Acoustics, Speech and Signal Processing*, pp. 4580–4584, 2015.

[62] L. Deng and J. C. Platt, "Ensemble deep learning for speech recognition," in *Proc. Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[63] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning.* MIT Press Cambridge, 2016, vol. 1.

[64] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[65] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Distributed deep learning models for wireless signal classification with low-cost spectrum sensors," *CoRR, vol. abs/1707.08908*, 2017.