LightAMC: Lightweight Automatic Modulation Classification via Deep Learning and Compressive Sensing

Yu Wang [1], Jie Yang [1], and Miao Liu [1]

[1]Affiliation not available

October 30, 2023

Abstract

Automatic modulation classification (AMC) is an promising technology for non-cooperative communication systems in both military and civilian scenarios. Recently, deep learning (DL) based AMC methods have been proposed with outstanding performances. However, both high computing cost and large model sizes are the biggest hinders for deployment of the conventional DL based methods, particularly in the application of internet-of-things (IoT) networks and unmanned aerial vehicle (UAV)-aided systems. In this correspondence, a novel DL based lightweight AMC (LightAMC) method is proposed with smaller model sizes and faster computational speed. We first introduce a scaling factor for each neuron in convolutional neural network (CNN) and enforce scaling factors sparsity via compressive sensing. It can give an assist to screen out redundant neurons and then these neurons are pruned. Experimental results show that the proposed LightAMC method can effectively reduce model sizes and accelerate computation with the slight performance loss.

LightAMC: Lightweight Automatic Modulation Classification via Deep Learning and Compressive Sensing

Yu Wang, *Student Member*, *IEEE*, Jie Yang, *Member*, *IEEE*, Miao Liu, *Member*, *IEEE*, and Guan Gui, *Senior Member*, *IEEE*

***Abstract*—Automatic modulation classification (AMC) is an promising technology for non-cooperative communication systems in both military and civilian scenarios. Recently, deep learning (DL) based AMC methods have been proposed with outstanding performances. However, both high computing cost and large model sizes are the biggest hinders for deployment of the conventional DL based methods, particularly in the application of internet-of-things (IoT) networks and unmanned aerial vehicle (UAV)-aided systems. In this correspondence, a novel DL based lightweight AMC (LightAMC) method is proposed with smaller model sizes and faster computational speed. We first introduce a scaling factor for each neuron in convolutional neural network (CNN) and enforce scaling factors sparsity via compressive sensing. It can give an assist to screen out redundant neurons and then these neurons are pruned. Experimental results show that the proposed LightAMC method can effectively reduce model sizes and accelerate computation with the slight performance loss.**

***Index Terms*—Lightweight automatic modulation classification (LightAMC), convolutional neural network (CNN), neuron pruning, compressive sensing.**

I. Introduction

Automatic modulation classification (AMC) is a novel and promising technology in the non-cooperative communication scenarios with neither agreement nor authorization between the transmitters and the receivers. The AMC methods have various applications in military and civilian fields, including intercepted signal recovering or spectrum monitoring [1]–[3]. There are two typical AMC methods based on features and likelihood functions, respectively. This paper focuses on the feature-based AMC method, which can be modeled as a pattern recognition problem based on extracted features without any prior information. Typical manmade features include high order statistical features, wavelet transform-based features and so on.

In recent years, deep learning (DL) has been considered one of the most effective tools to solve various problems in wireless communications [4]–[14], because of its powerful feature extraction capability, especially confronting with high-dimensional data. At the same time, many DL-based AMC methods were proposed for the higher classification performances. T. O'Shea *et al*. proposed a convolutional neural network (CNN)-based AMC method using the in-phase and quadrature (IQ) component of signals [15]. S. Rajendran *et al*. proposed a novel data-driven automatic modulation classification based on the long short term memory (LSTM) [16]. What' more, S. Hu, *et al*. [17] demonstrated the robustness of DL-based AMC method in more complex noise conditions, including white non-Gaussian noise and time-correlated non-Gaussian noise. However, DL models generally have large model sizes and slow computation speed. Thus, it is difficult to apply these methods into the edge devices [18]–[21], such as internet-of-thing (IoT) devices and unmanned aerial vehicle (UAV), which are equipped with the limited device memories and computation capability. Hence, the compression and acceleration of DL models are necessary in the future development of the DL-based wireless communications.

In this paper, we propose a compressive sensing (CS)-based neuron pruning technology to implement the lightweight AMC (LightAMC) method. Specifically, the proposed method is implemented by pruning redundant neurons via ℓ_1 regularization on the fundament of the mixed dataset-based AMC (M-AMC) method. Experimental results are given to confirm the proposed method in terms of the model sizes, the computation time as well as the classification performance.

II. Signal Model and Problem Formulation

Consider that the received signal $y = \{y(k)\}_{k=1}^{K}$ is a complex-valued baseband signal, and its sampling strictly follows Nyquist criterion. The signal in the k-th sampling moment can be expressed by

$$y(k) = Ae^{j\varphi}s(k) + w(k), \tag{1}$$

where $s(k)$ is a modulation signal, and the energy of modulation signal is normalized for the fair classification of different modulation types, i.e., $\sum_{k=1}^{K}|s(k)|^2 = 1$; $w(k)$ is zero-mean additive white Gaussian noise (AWGN). In addition, A and φ are the channel gain and the phase offset, respectively. These two parameters are constant under the assumption of the flat fading and time-invariant channel.

This work was supported by the Project Funded by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant TC190A3WZ-2, the Jiangsu Specially Appointed Professor under Grant RK002STP16001, the Innovation and Entrepreneurship of Jiangsu High-level Talent under Grant CZ0010617002, the Six Top Talents Program of Jiangsu under Grant XYDXX-010, the 1311 Talent Plan of Nanjing University of Posts and Telecommunications. *(Corresponding author: Guan Gui.)*

The authors are with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (E-mails: {1018010407, jyang, liumiao, guiguan}@njupt.edu.cn).

Based on the signal model, two independent datasets are prepared for the training and testing of the CNN. The sample in the datasets consists of the in-phase and quadrature (IQ) component of the received signal. The IQ component also represents real part and imaginary part of the received signal, respectively. Thus, $I = \{\text{real}[y(k)]\}_{k=1}^{K}$, and $Q = \{\text{imag}[y(k)]\}_{k=1}^{K}$. Then, the IQ sample can be written as

$$IQ = \begin{Bmatrix} \text{real}[y(k)] \\ \text{imag}[y(k)] \end{Bmatrix}_{k=1}^{K}, \tag{2}$$

and it is a $2 \times K$ real-valued matrix. Then, we specifically describe the problem of AMC. AMC is a technique to blindly identify modulation types in the range of a limited modulation type candidate pool $\Theta = \{\theta_i\}_{i=1}^{M}$, where θ_i is one of the modulation types. According to the maximum-a-posterior (MAP) criterion, this problem can be described as

$$\theta_i = \underset{\theta_i \in \Theta}{\arg\max} \ P(\theta_i|y), \tag{3}$$

where $P(\theta_i|y)$ is the probability distribution function (PDF) of θ_i under the condition of the received signal y. Without the loss of generality, we consider two modulation type candidate pools, i. e., $\Theta_1 = \{\text{BPSK, QPSK, 8PSK}\}$ [22], [23] and $\Theta_2 = \{\text{BPSK, QPSK, 8PSK, 16QAM}\}$ [1], [17].

III. Existing AMC Methods

A. Traditional AMC Method Based on HOC and SVM

The architecture of the traditional method is depicted in Fig. 1. It consists of pre-processing, feature extraction, classifier, and SNR estimation. Pre-processing contains signal conversion, frequency and phase synchronization and so on. Feature extraction is the core step of the AMC method and we consider the fourth order cumulants as features [22]. For the received baseband complex signal $y = \{y(k)\}_{k=1}^{K}$, the i-th order moment is defined as $M_{ij} = E[y^{i-j}y^{*j}]$. Then, the fourth-order cumulants [22] are defined as

$$C_{40} = M_{40} - 3M_{20}^2, \tag{4}$$

$$C_{41} = M_{41} - 3M_{21}M_{20}, \tag{5}$$

$$C_{42} = M_{42} - |M_{20}|^2 - 2M_{21}^2. \tag{6}$$

In the actual applications, the number of the sampling points for the received signal y is limited. Hence, the estimation value of i-th order moments $\hat{M}_{ij} = \frac{1}{K}\sum_{k=1}^{K}[y^{i-j}(k)y^{*j}(k)]$ is applied to replace the theoretical value. Similarly, the estimation value of fourth-order cumulants $\hat{C}_{4j}$, $j \in \{0, 1, 2\}$ can be obtained by replacing M_{ij} in (4–6) with $\hat{M}_{ij}$. Considering the scale problem, the estimation value of fourth-order cumulants is generally normalized, and the normalized value of fourth-order cumulants [23] is

$$\widetilde{C}_{4j} = \frac{\hat{C}_{4j}}{\frac{1}{K}(\sum_{k=1}^{K}|y(k)|^2)^2}, \text{where } j \in \{0, 1, 2\}, \tag{7}$$

and $\{\widetilde{C}_{4j}\}_{j=0}^{2}$ is a feature vector for training SVM to classify different modulation signals. In addition, SVM is applied for the module "Classification", and multiple SVM models must be prepared for the varying SNR conditions. What's more, it is necessary to estimate the real-time SNR via the module "SNR estimation" to choose the suitable SVM model from multiple SVM models for the correct classification.

TABLE I
THE STRUCTURE OF CNN, INCLUDING LAYER NUMBER, LAYER TYPE AND LAYER STRUCTURE.

NO.	Type	Structure
-	-	Input (IQ samples, labels)
1	Conv	Conv2D (u_1, 1×8) + BN + ReLU + Dropout (0.05)
2	Conv	Conv2D (u_2, 2×4) + BN + ReLU + Dropout (0.05)
3	FC	Dense (u_3) + BN + ReLU + Dropout (0.5)
4	FC	Dense (u_4) + BN + ReLU + Dropout (0.5)
5	FC	Dense (M) + Softmax

Tips: "Conv" represents the convolutional layer and "FC" is the fully-connected layer.

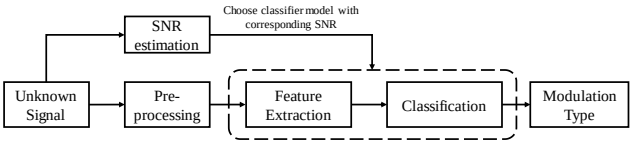


Fig. 1. The architecture of feature-based AMC method. The key steps of AMC method are feature extraction and classification. In the traditional AMC method, HOC is applied to characterize the difference of various modulation types, while SVM is as a classifier. In the F-AMC method, CNN is both the feature extractor and classifier. What's more, SNR estimation gives an assist to choose corresponding SVMs or CNNs, because SVM or CNN in these previously proposed methods is trained on the IQ samples with the fixed SNR, and they can just be adopted into the corresponding SNR.

B. CNN-based F-AMC Methods

CNN-based AMC methods have the similar architecture with the traditional AMC method, and the only difference is that HOC and SVM is replaced by CNN. Existing CNN-based AMC methods consider the training CNN model on fixed samples with single SNR. Here, it is referred as the F-AMC method. The architecture of the F-AMC method is shown in Fig. 1, where CNN is applied to replace the modules of "Feature Extraction" and "Classification".

IV. The Proposed LightAMC Method

Our proposed CNN-based LightAMC method is introduced in this section. A novel neuron pruning technology based on ℓ_1 regularization is applied to realize the LightAMC method. The implementation of the LightAMC method consists of training, pruning and finetuning.

A. CNN Training for M-AMC Method

Training is to train a CNN, designed by the artificial experiences, to classify the modulation signals. In the F-AMC methods [1], [15], [17], CNN is trained on the fixed SNR, and they are only useful for the IQ samples with the corresponding SNR. It means that the SNR estimation is necessary for the choice of the suitable CNN model, which is shown in Fig. 1. In addition, the F-AMC methods have the weak robustness under varying noise condition.

Unlike the F-AMC methods, the M-AMC method is proposed with the capability to confront with the varying noise condition without the assist of the SNR estimation. Because M-AMC method is trained on the mixed datasets with multiple SNRs, and the CNN can extract more robust and general features from mixed dataset. Assuming that SNR is from -10 dB to 10 dB, we use multiple IQ samples with different SNRs (it ranges from -10 dB to 10 dB with 2 dB as an interval) and mix them equally to create the mixed dataset for training.

In addition, for the fair comparison of the F-AMC and M-AMC method, we consider to apply the same CNN structure into both F-AMC and M-AMC method, which is shown in Tab. I. This CNN has five layers, containing two convolutional layers and three fully-connected layers. In first four layers, the numbers of neurons are denoted as $\{u^l\}_{l=1}^4$, which is equal to $\{128, 64, 256, 128\}$, and rectified linear unit ($ReLU$) $f_{ReLU}(\cdot) = max(\cdot, 0)$ is as the activation function for these four layers. The number of neurons in the last layer is decided by the number of modulation types M, and $Softmax$ is the activation function for the last layer.

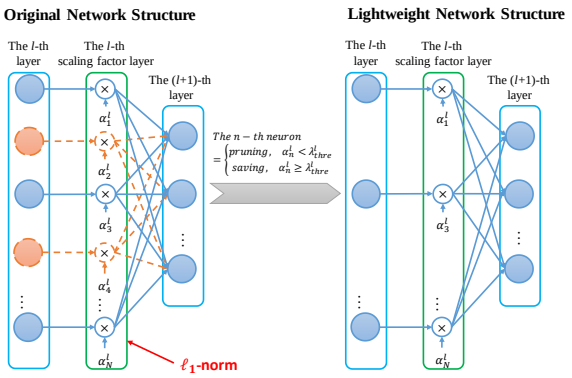


Fig. 2. The principle of neuron pruning based on ℓ_1 regularization.

B. Neuron Pruning via ℓ_1 Regularization

Pruning is to cut down unimportant neurons, and how to measure the importance of neurons is critical. Inspired by [24] [25], we introduce a learnable metrics $\alpha = \{\alpha^l\}_{l=1}^4$ based on ℓ_1 regularization for this problem. α represents the scaling factor, and it is multiplied by the output of each neuron in the first four layers of CNN. Hence, the output of one of the first four layers in CNN can be represented by

$$x^{l+1} = \alpha^l \cdot \max\{\gamma^l \cdot BN_{\mu^l,\sigma^l,\epsilon^l}(W^l * x^l) + \beta^l, 0\}, \quad (8)$$

where x^l and x^{l+1} is the input and output, and W^l, γ^l and β^l are the trainable weight. In addition, $BN_{\mu^l,\sigma^l,\epsilon^l}(z_{in}) = \frac{z_{in}-\mu^l}{\sqrt{(\sigma^l)^2+\epsilon^l}}$, where $\mu^l = E(z_{in})$, $(\sigma^l)^2 = Var(z_{in})$ and ϵ^l is a minimum value to prevent that σ^l is zero.

Then, we jointly train trainable parameters: $W = \{W^l\}_{l=1}^5$, $\gamma = \{\gamma^l\}_{l=1}^4$ and $\beta = \{\beta^l\}_{l=1}^4$, and scaling factor $\alpha = \{\alpha^l\}_{l=1}^4$ with sparsity constraint. Finally, we prune the neurons, whose scaling factor is smaller than threshold λ_{thre}. please notice that $\lambda_{thre} = \{\lambda_{thre}^l\}_{l=1}^4$ is layered, and we usually set the 80% of the average value of all elements as the threshold [24], but the detailed threshold is chosen by many trying and testing on the validation samples as few neurons and as low validation performance loss as possible. The principle of neuron pruning is shown in Fig. 2. It is noted that the loss function is different and it is give as

$$\arg\max_{W,\gamma,\beta,\alpha} \sum_{(x,y)} l(f(x; W, \gamma, \beta), y) + \lambda \sum_{l=1}^{4} ||\alpha^l||_1, \quad (9)$$

where (x, y) is training sample and label. The first term in (9) represents experience loss and we apply cross entropy loss function as this term. The second term is sparsity-inducing penalty, and we choose ℓ_1 regularization, which is widely applied into achieve sparsity [26]. In addition, λ is utilized to balance the two terms.

We train a CNN with $\lambda \in \{0.001, 0.005, 0.01\}$ for M-AMC method on Θ_1, and the distribution of α^3 (it is the scaling factor following behind the first fully-connected layer) is shown in Fig. 3. It is obvious that the larger λ generally leads to the higher sparsity. However, too large λ will cause instability of the algorithm (9). Correct classification probability of $\lambda = 0.005$ and 0.001 can reach up to nearly 100% at SNR = 10 dB, while that of $\lambda = 0.01$ only has 67%.

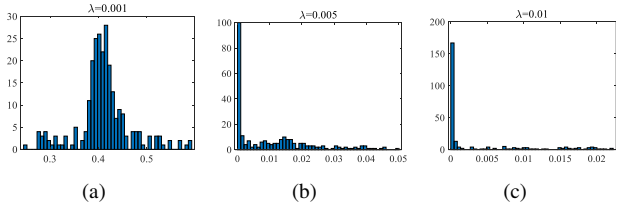


Fig. 3. The distribution of α^3 in the CNNs of the M-AMC method under the different values of λ, which are trained on θ_1. The values of λ are equal to 0.001, 0.005 and 0.01 in (a), (b) and (c), respectively. With the increase of the value of λ, the sparse degree is increasing gradually, but when the value of λ is equal to 0.01, the correct classification probability at the high SNR is greatly decreased.

C. Finetuning in Short Epochs

Although the scaling factor with ℓ_1 regularization can give an assist for us to correctly distinguish the unimportant neurons, it is inevitable to bring the performance loss with neuron pruning technology. Thus, finetuning is a critical step to recover the classification performance by retraining the trimmed CNN. Specifically, finetuning is to retrain the trimmed CNN in the short epochs, after loading the weights of saved neurons and setting λ=0. Hence, the performance loss is limited with the step of finetuning.

V. Experimental Results

In this section, the effectiveness of the LightAMC method is empirically demonstrated on Θ_1 and Θ_2, which are independent from the training datasets and contains 6000 IQ samples per modulation type per SNR. Our proposed LightAMC method is evaluated in terms of three aspects of model size, computation time and classification performance.

Algorithm 1 The proposed LightAMC method.

Input: IQ samples with SNRs ranging from -10 dB to 10 dB with 1 dB as an interval, and 6000 IQ samples per modulation type per SNR;
Output: The CNN for LightAMC;
1: Choose IQ samples with SNR $= \{-10, -8, \cdots, 8, 10\}$ dB and mix these samples proportionally;
2: Divide randomly the mixed samples into training samples and validation samples by $7:3$;
3: Construct CNN according to Tab. I and introduce scaling factor $\{\alpha^l\}_{l=1}^4$ multiplied by the output of each neurons in the first four layers in CNN;
4: Set a proper λ ($\lambda = 0.005$ for Θ_1 and $\lambda = 0.006$ for Θ_2), choose stochastic gradient descent (SGD) as optimizer and train CNN to minimize loss function (9);
5: Set a proper threshold λ_{thre} for each layer, and cut down neurons, whose corresponding scaling factor is smaller than threshold;
6: Set five retraining epochs, load the weights of the unpruned neurons and retrain the trimmed CNN on training samples to recover its performance;
7: **return** CNN.

TABLE II
STRUCTURES AND MODEL SIZES OF THREE CNN-BASED METHODS IN Θ_1.

Method	$\{u_l\}_{l=1}^4$	Model size (MB)
F-AMC	{128,64,256,128}	15.5×21
M-AMC	{128,64,256,128}	15.5
Traditional AMC	-	-
LightAMC (Proposed)	{77,18,49,44}	1.0 (93.5% ↓)

A. Model Size and Device Memory

The most obvious improvement of the LightAMC method is that it has smaller sizes and requires fewer device memories than F-AMC method, which is shown in Tabs. II–III. The improvement is beneficial from not only the application of M-AMC method but also the implementation of neuron pruning technology. When SNR ranges from -10 dB to 10 dB with 1 dB as an interval, we need to train twenty-one CNN models for F-AMC method, but only one CNN model is required in M-AMC method, benefitting from the application of mixed datasets. So, considering that the same CNN structure is applied into F-AMC and M-AMC method, CNN model sizes are same, but required device memories for M-AMC are only $1/21$ of that for F-AMC method.

One the basis of M-AMC method, LightAMC method is to prune unimportant neurons of CNN. The trimmed CNN structure is shown in Tabs. II–III, and the number of neurons for each layer, especially two fully-connected layers, has been reduced greatly. With the reduction of neurons, the CNN model sizes and the required device memories are also declining sharply. Specifically, CNN model sizes of LightAMC method only has 6.5% of that of M-AMC method in Θ_1, and the ratio is 8.4% in Θ_2. In addition, we do not compare CNN-based AMC method with traditional AMC method, because the SVM in traditional AMC method is "Scikit-learn" as backend and the CNN is based on "Keras". Hence, it is meaningless to compare these two kinds of AMC methods in the model sizes.

TABLE III
STRUCTURES AND MODEL SIZES OF THREE CNN-BASED METHODS IN Θ_2.

Method	$\{u_i\}_{l=1}^4$	Model size (MB)
F-AMC	{128,64,256,128}	15.5×21
M-AMC	{128,64,256,128}	15.5
Traditional AMC	-	-
LightAMC (Proposed)	{81, 19, 63, 49}	1.3 (91.6% ↓)

B. Computation Time

The computation time of different AMC methods is depicted in Fig. 4. For the fair comparison, we test the computation time on the same device equipped with i7-8750H and GTX1080Ti, and coding is based on python. It can be observed that the LightAMC method is faster than F-AMC and M-AMC method, because pruning the redundant neurons leads to the reduction of the redundant computation. Since F-AMC and M-AMC method have the same structure, hence the computation time of which are 44.2 μs per sample. While the computation time of the LightAMC method decreases by almost 24%, which is 33.2 μs per sample in Θ_1 and 33.6 μs per sample in Θ_2. Unlike these CNN-based methods, it is difficult to run traditional AMC method on GPU. Hence, the computation time of traditional AMC method is running on CPU, and it is obviously slower than these CNN-based methods.

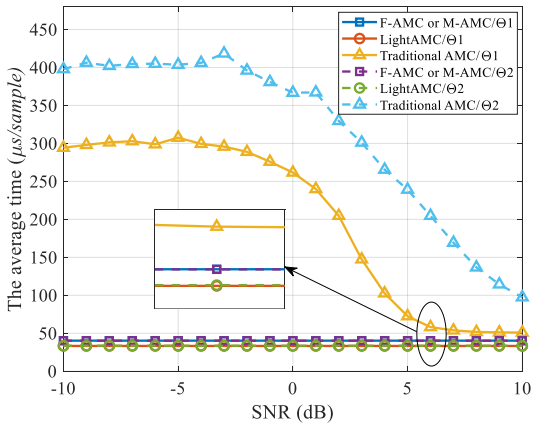


Fig. 4. The average computation time of per sample in the different AMC methods.

C. Classification Performance

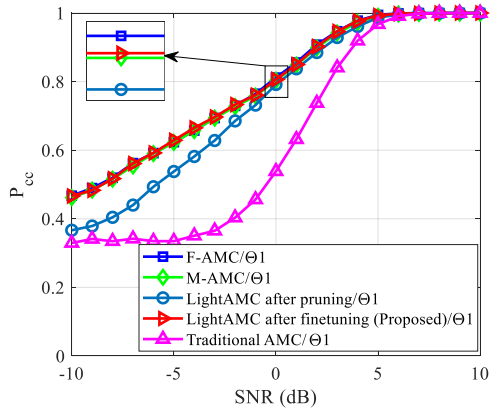
Correct classification probability ($P_{cc} = \{P_{cc}^i\}_{i=-10}^{10}$) is applied to measure performance and it is given as

$$P_{cc}^i = \frac{S_{correct}^i}{S_{total}}, \tag{10}$$

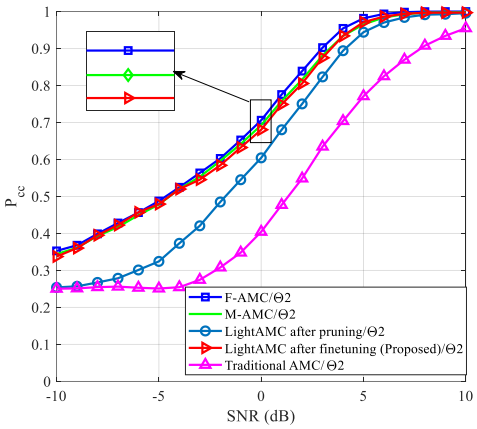
where $S_{correct}^i$ is the number of correctly classified samples under SNR $= i$ dB and S_{total} is 18000 in Θ_1 or 24000 in Θ_2. The curve of P_{cc} is shown in Fig. 5.

While compressing model size and reducing the computation time, the pruned LightAMC method has almost 10% performance gap with other CNN-based AMC methods, and its performance is still higher than the traditional AMC method. However, after a short term finetuning, the gap is extremely

reduced. The finetuned LightAMC method has the similar performance with the M-AMC method, and it has the limited performance loss, compared with the F-AMC method.



(a)



(b)

Fig. 5. The classification performance of different AMC methods. (a) is the performance in Θ_1, (b) is the performance in Θ_2.

VI. Concluding Remarks

In this correspondence, we have proposed an effective LightAMC method using CNN and compressive sensing under varying noise regimes. The proposed LightAMC method is based on the M-AMC method that is trained on mixed dataset with multiple SNRs. Then, ℓ_1 regularization-based neuron pruning technology is applied to cut down redundant neurons in the CNN for the M-AMC method. Hence, the proposed LightAMC method requires fewer device memories and has faster computational speed under the limited performance loss. Simulation results also demonstrate the superiority of the LightAMC method. In the future work, we will consider iterative shrinkage-thresholding algorithm (ISTA) as the optimizer to replace SGD for LightAMC method with ℓ_1 regularization, and we expect that ISTA-based LightAMC method can achieve the better performance.

References

[1] F. Meng, et al., "Automatic modulation classification: A deep learning-enabled approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10760–10772, 2018.

[2] I. Bisio, C. Garibotto, F. Lavagetto, and A. Sciarrone, "Outdoor places of interest recognition using WiFi fingerprints," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5076-5086, 2019.

[3] I. Bisio, C. Garibotto, F. Lavagetto, A. Sciarrone , S. Zappatore, "Blind detection: Advanced techniques for WiFi-based drone surveillance," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 938–946, 2019.

[4] Z. Md. Fadlullah, et al., "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Commun. Surveys and Tuts.*, vol. 19, no. 4, pp. 2432-2455, 2017.

[5] N. Kato, et al., "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," *IEEE Wirel. Commun. Mag.*, vol. 24, no. 3, pp. 146–153, 2016.

[6] G. Gui, et al., "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, 2018.

[7] H. Huang, et al., "Deep learning for physical-Layer 5G wireless techniques: Opportunities, challenges and solutions," *IEEE Wirel. Commun.*, to be published, doi: 10.1109/MWC.2019.1900027.

[8] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, D. Zhao, "Flight delay prediction based on aviation big data and machine learning," *IEEE Trans. Veh. Technol.*, doi: 10.1109/TVT.2019.2954094.

[9] H. Huang, et al., "Fast beamforming design via deep learning," *IEEE Trans. Veh. Technol.*, to be published, doi: 10.1109/TVT.2019.2949122.

[10] Z. Md. Fadlullah, et al., "On intelligent traffic control for large-scale heterogeneous networks: A value matrix-based deep learning approach," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2479–2482, 2018.

[11] F. Tang, et al., "An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A deep learning approach," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5141–5154, 2018.

[12] H. Huang, et al., "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, 2019.

[13] Z. Md. Fadlullah, et al., "Value iteration architecture based deep learning for intelligent routing exploiting heterogeneous computing platforms," *IEEE Trans. Computers*, vol. 68, no. 6, pp. 939–950, 2019.

[14] Y. Wang, et al., "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, 2019.

[15] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, 2017.

[16] S. Rajendran et al., "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, 2018.

[17] S. Hu, et al., "Deep neural network for robust modulation classification under uncertain noise conditions," *IEEE Trans. Veh. Technol.*, to be published, doi: 10.1109/TVT.2019.2951594.

[18] Z. Zhou, et al., "Access control and resource allocation for M2M communications in industrial automation," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 3093–3103, 2019.

[19] M. Liu, et al., "Deep cognitive perspective: resource allocation for NOMA based heterogeneous IoT with imperfect SIC," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2885–2894, 2019.

[20] Z. Zhou, et al., "Energy-efficient resource allocation for D2D communications underlaying cloud-RAN-based LTE-A networks," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 428–438, 2016.

[21] F. Tang, et al., "AC-POCA: Anti-coordination game based partially overlapping channels assignment in combined UAV and D2D based networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1672–1683, 2018.

[22] A. Swami, B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Trans. Commun.*, vol. 48, no. 3, pp. 416–429, 2000.

[23] K. Hassan, et al.,"Blind digital modulation identification for spatially-correlated MIMO systems," *IEEE Trans. Wirel. Commun.*, vol. 11, no. 2, pp. 683–693, 2011.

[24] Z. Liu, et al., "Learning efficient convolutional networks through network slimming," in *ICCV*, Venice, Italy, Oct. 22–29, 2017, pp. 2736–2744.

[25] J. Ye, et al., "Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers," in *ICLR*, Vancouver, Canada, Apr. 30–May 3, 2018, pp. 1–11.

[26] Y. Li, et al., "Nonconvex penalized regularization for robust sparse recovery in the presence of $S\alpha S$ noise," *IEEE Access*, vol. 6, no. 1, pp. 25474–25485, 2018.