A Deep Learning Method Based on Convolution Neural Network for Blind Demodulation of Mixed Signals with Different Modulation Types

Hongtao Zhu[1], Zhenyong Wang[1,2](✉), Dezhi Li[1], Qing Guo[1],
and Zhenbang Wang[1]

[1] School of Electronics and Information Engineering,
Harbin Institute of Technology, Harbin, Heilongjiang, China
zhuhongtao@stu.hit.edu.cn,
{ZYWang,lidezhi,QGuo}@hit.edu.cn, zhenbangw@163.com
[2] Shenzhen Academy of Aerospace Technology, Shenzhen, Guangdong, China

Abstract. In recent years, deep learning is becoming more and more popular. It has been widely used in image recognition, automatic speech recognition and natural language processing. In the field of communication, the signal is considered as time data, which can identify the intrinsic characteristics and information of the signal by the way of deep learning. In the aspect of cognitive radio, if the signal adopts different modulation methods in different time slots for adaptive modulation, it is difficult for the existing signal demodulation system to demodulate it effectively. Usually, it is necessary to identify the modulation mode of the signal first. In this context, the deep learning is introduced into signal demodulation. On the basis of analyzing the structure of convolutional neural network (CNN), an improved CNN structure is proposed, which does not need to recognize modulation methods and realizes blind demodulation of mixed signals with different signal-to-noise ratio (SNR). Through transfer learning and denoising auto-encoder, the network is further optimized to further reduce the bit error rate (BER).

Keywords: Signal demodulation · Convolution neural network · Transfer learning · Denoising auto-encoder

1 Introduction

With the development of cognitive radio, it becomes more and more important to identify the modulation mode and demodulate the signal. For general signal demodulation process, first of all, it is necessary to identify the modulation mode of the signal, so a series of algorithms are needed to estimate the parameters, and then the signal is demodulated when the modulation mode and other synchronization parameters are known. In order to adapt to different hardware and communication environments, different modulation modes, such as PSK, QAM, FSK, are usually needed when transmitting signals. Considering such a situation, if the transmitter modulates the

M. Jia et al. (Eds.): WiSATS 2019, LNICST 280, pp. 91–103, 2019.
https://doi.org/10.1007/978-3-030-19153-5_9

signal randomly in different time slots by different modulation methods, it can greatly increase the degree of confidentiality of the information, and also can choose the modulation mode independently according to the actual requirements in transmitting the information. Thus, if the modulation classification and other technical communications are not the same. The modulation of the signal is recognized, and the existing system is unable to demodulate the signal effectively.

Based on the above problems, how to realize the unified demodulation of mixed signals with multiple modulation modes without changing the hardware circuit is a challenging research problem. In recent years, with the development of computer hardware and the application of distributed computing, it is possible to train large-scale neural networks. More and more people are devoting themselves to the study of deep learning. Deep learning is a nonlinear model that captures and learns the abstract representation of data. It can be used in one layer with neural networks. A large number of neurons adapt to [1]. Abstract learning in depth learning can be understood as a demodulation process, which means extracting useful information from the original data [2]. In signal demodulation, Milad and Azizm adopt ANN demodulation quadrature amplitude modulation (QAM) [3]. Compared with general QAM demodulator, this method is helpful to reduce the bit error rate. In reference [4], a deep neural network (DNN) and a long-term and short-term memory (LSTM) for FM demodulation are proposed. It uses a learning-based method to use the priori information of the voice message sent in the demodulation process. Their reconstruction depends on the use of memory, so their demodulation schemes may fail without strong memory. Mira proposed an adaptive radial basis function (RBF) neural network in reference [5] for multiuser communication demodulation in direct sequence spread spectrum multiple access (DSSSMA). The results show that it has fast convergence and near optimal performance. Akan and Dogan proposed a NN-based software radio receiver in the literature [6] considering the unknown channel model of training sequence. The receiver has the same performance as the correlation receiver. It is suitable for additive white Gaussian noise (AWGN) channel and has the advantages of processing interference and phase offset channels. In addition, the receiver can also be used for fading channels with improved performance.

The main idea of this paper is to propose a neural network model based on CNN, which can demodulate a variety of mixed modulation signals in a unified way. The rest of this paper is arranged as follows: Sect. 2 studies the basic principle of CNN and analyzes the characteristics of CNN for signal demodulation. Section 3 presents a signal demodulation model suitable for various mixed modulation signals. Based on this model, the model is improved by using transfer learning and denoising auto-encoder. Section 4 compares the demodulation performance of the improved model with that of the original model through experiments, and verifies that the network model can demodulate the mixed signals with different SNR. Section 5 conclude the paper.

2 The Principle of Convolutional Neural Network

CNN is different from general neural network in that it is formed by stacking convolution layer and pool layer. The convolution layer of convolution neural network usually contains three levels. In the first stage, the convolution layer uses multiple convolution kernels in parallel to generate a set of linear activation functions. In the second stage, nonlinear activation functions such as rectifier linear element functions act on each linear output at the first stage. In the third level, the pooling function is used to further adjust the output of the volume layer.

The convolution nucleus can be seen as a layer of neuronal structure, which can also weigh inputs and produce output signals. The characteristic graph is the output of a convolution kernel. A given filter is dragged across the entire previous feature map or input, and each move of a step produces activation of neurons at each location and outputs to a feature map. From the point of view of communication, each convolution kernel is equivalent to a filter, which can extract the corresponding data features.

2.1 Basic Convolution Function

Convolution operations used in neural networks are usually not the standard discrete convolution operations used in mathematical literature, and they are slightly different from them. Former usually refers to a specific operation, which involves the parallel use of multiple convolutions. Convolution with a single kernel can only extract one type of feature, although it acts on multiple spatial locations. But we often hope that one layer of neural network can extract many types of features in multiple locations.

Suppose that there is a four-dimensional kernel tensor K, each of which is $K_{i,j,k,l}$, indicating the connection strength between an output unit in channel i and an input unit in channel j, and there is a k-row l-column offset between the output unit and the input unit. Suppose the input is composed of observation data V, each element being $V_{i,j,k}$, representing the value in column k and row j in channel i. Assuming that the output Z and the output V have the same form, if the output Z is obtained by convoluting K and V, and it is given by:

$$Z_{i,j,k} = \sum_{l,m,n} V_{l,j+m-1,k+n-1} K_{i,l,m,n} \tag{1}$$

Sometimes it is desirable to reduce the overhead of network computing by downsampling. If you want to sample only s pixels in each direction of the output, then you can define a downsampling convolution function C as shown in Eq. (2).

$$Z_{i,j,k} = c(K, V, s)_{i,j,k} = \sum_{l,m,n} \left[V_{l,(j-1)\times s+m,(k-1)\times s+n} K_{i,l,m,n}\right] \tag{2}$$

Where s is called the step size of downsampling. Figure 1 is a convolution with step size.

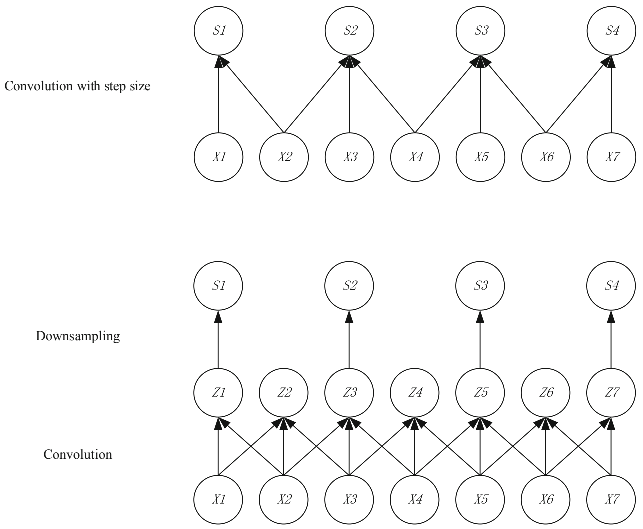

Fig. 1. Convolution with step size

2.2 Characteristics of CNN

Convolution operation helps to improve the model through two important ideas: sparse interactions and parameter sharing [7].

Sparse Interactions. Each output unit of traditional neural network interacts with each input unit. However, CNN has the characteristics of sparse interaction. By using kernel function, the parameters of model storage are greatly reduced. For example, when detecting and recognizing signals, the input signal sample points may contain millions of sample points, but small meaningful features such as frequency hopping can be detected by occupying only a few to hundreds of core signal sample points. This means that fewer parameters need to be stored, which not only reduces the storage requirements of the model, but also improves its statistical efficiency. Graphical explanations of sparse connections are shown in Fig. 2.

Parameter Sharing. Parameter sharing is to use the same parameters in multiple functions of a model. In traditional feedforward neural networks, each element of the weight matrix is used only once when calculating the output of a layer. In convolution neural network, the convolution kernel will do convolution operation with each position of input, and optimize the network by learning a set of parameters, thus greatly reducing the network parameters. Figure 3 shows how the parameter sharing is implemented.

2.3 Optimization Method and Selection of Activation Function

Optimization plays a crucial role in deep learning algorithms. Optimization refers to minimizing or maximizing a task function $f(x)$ by changing x, which is called a cost

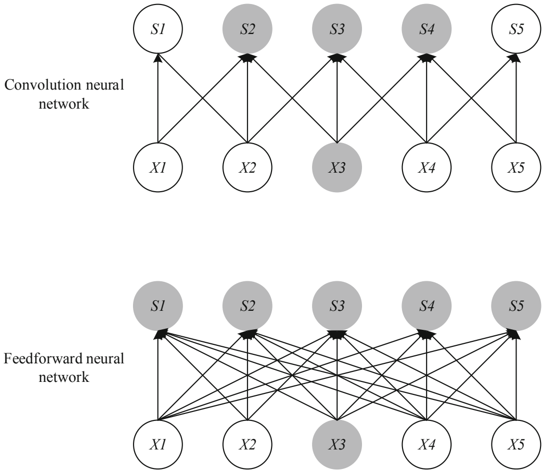

Fig. 2. Comparison of connections (above is a convolution network, the following is a traditional feedforward neural network).

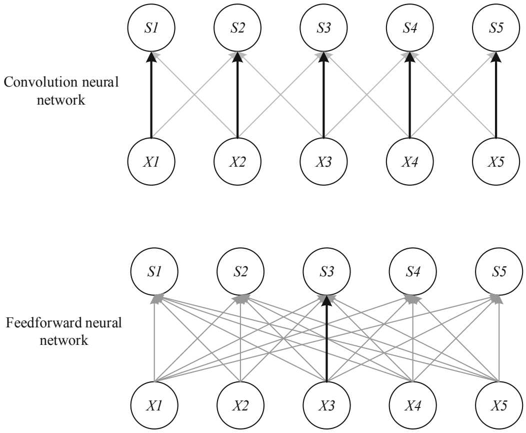

Fig. 3. Parameter sharing (above is a convolution network, the following is a traditional feedforward neural network).

function. How to choose the cost function is an important link in the design of neural network. The choice of error function will affect the convergence of the model, optimization speed and other aspects. In this paper, the mean square error function is chosen as the cost function to train.

Considering a set of data sets $X = \{x^{(1)}, \ldots, x^{(m)}\}$ with m samples and corresponding data sets labeled $Y = \{y^{(1)}, \ldots, y^{(m)}\}$, the error function can be expressed as follows:

$$\theta_{MSE} = \arg\min_{\theta} \prod_{i=1}^{m} E(\bar{y}_i - y_i)^2 \tag{3}$$

If $\bar{y}$ is an unbiased estimate of y, then the estimator evaluated by mean square error is consistent with the estimator evaluated by variance, so the unbiased estimator can be examined by variance. When $\bar{y}$ is not an unbiased estimate of y, the two are no longer equivalent, then the mean square error is reasonable, that is, through the variance and deviation of two parts of the judgment.

Suppose there is a function $y = f(x)$ in which x and y are real numbers. The derivative of this function is $f'(x)$. The derivative $f'(x)$ represents the slope of $f(x)$ at point x. The size of the slope shows how to input the size of the change to the desired output $f(x+\varepsilon) \approx f(x) + \varepsilon f'(x)$.

The derivative is useful for minimizing a function because it specifies how to change x to improve y slightly. So x can move $f(x)$ to a smaller step in the opposite direction of the derivative to reduce the amount of it. Stochastic Gradient Descent (SGD) is a common algorithm used to train neural networks. The loss function in the deep learning algorithm is usually the sum of the loss functions of each sample. For example, the negative conditional logarithmic likelihood of training data can be written as Eq. (4).

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L(x^{(i)}, y^{(i)}, \theta) \tag{4}$$

Where L is the loss function for each sample, and it can be expressed as Eq. (5).

$$L(x, y, \theta) = -\log p(y|x; \theta) \tag{5}$$

For these additive loss functions, gradient descent requires the following formula.

$$\nabla_{\theta} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \nabla_{\theta} L(x^{(i)}, y^{(i)}, \theta) \tag{6}$$

Considering the computational complexity and the computational efficiency, a general approach is to randomly and evenly extract a small number of samples from the training set called minibatch $B = \{x^{(1)}, \ldots, x^{(m')}\}$. Here m' is the number of samples per batch, and the selection criterion is to select as few samples as possible under the premise of meeting the task requirements. The estimation of gradient can be expressed as follows.

$$g = \frac{1}{m'} \nabla_{\theta} \sum_{i=1}^{m'} L(x^{(i)}, y^{(i)}, \theta) \tag{7}$$

By using samples from B, the stochastic gradient descent algorithm uses the following gradient estimation:

$$\theta \leftarrow \theta - \varepsilon g \tag{8}$$

Where ε is the learning rate.

In this paper, the Sigmoid function is used as the final output layer function. Sigmoid has exponential function form and is a nonlinear action function. It is the closest biological neuron in physical sense, and it is also the most widely used activation function type. Neural network learning is based on a set of samples, including the input and output of the samples, which are one-to-one corresponding to the input and output neurons. The weights and thresholds of the neural network are initially arbitrarily initialized. The learning process is to adjust the weights and thresholds again and again by iteration to minimize the mean square error of the actual output and the expected output of the network. Given the input, a continuous output can be obtained based on Sigmoid function, and the output of Sigmoid function can also be expressed as probability, because the model finally decodes bit outputs 0 and 1. With this activation function, the output can be limited to the range of (0, 1) for subsequent threshold decoding. The form of the Sigmoid function is as follows:

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{9}$$

3 Demodulator Based on Convolutional Neural Network

3.1 Building a Signal Demodulation Model

Through the experimental analysis, it is found that reducing the total connection layer can greatly reduce the number of model parameters, and will not have a great impact on the accuracy. Based on the above analysis, the final network structure is shown in Fig. 4.

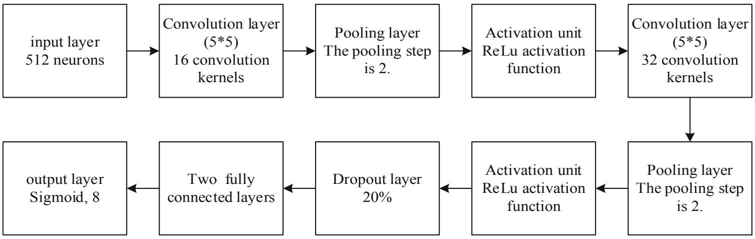

Fig. 4. Structure model of signal demodulation model based on convolution neural network.

In this paper, 8-bit information is used as a group, modulated to the carrier, each bit information is collected 64 points through the AWGN channel, so a group of 8-bit information is used for 512 sampling points as a sample input to the model. The number of input neurons is set to 512, which corresponds to 8 bit data of each sample and 512 sampling points. In this paper, there are two convolution layers in the neural network architecture, each layer is followed by a maximum pooling layer. The intermediate convolution layers are all modified linear elements. The output structure of the network is the Sigmoid unit of 8 neurons, which is used to output 8 bit demodulated bit data. The convolution kernel length is set to 5 * 5 size, and the convolution kernel number is 16 and 32 respectively. In addition, the maximum pooling step size is 2, and the width is set to 3. Because the network parameters are less, and random deactivation layer is added in the middle, the over-fitting phenomenon of the network can be effectively prevented.

In this paper, the mean square error between training data and model prediction is used as a cost function. For many tasks, even if small random noise is added to the input, the task should still be able to solve. However, the neural network proved to be not very robust to noise [8]. One of the ways to improve the robustness of neural networks is to simply apply random noise to its input and then train it. Therefore, in this experiment, when testing the accuracy of the algorithm under different SNR conditions, the signal data of different modulation modes under the SNR is collected first, 75% of the data is used as training set, and the rest as test set. The stochastic gradient descent algorithm is used as an optimization algorithm and the learning rate is set to 0.0001. Because the optimization algorithm is iterative, it needs to specify the initial value, here the Gaussian distribution initializer is used to randomly initialize the parameters.

3.2 Improved Demodulation Model

Transfer Learning. On Transfer learning refers to the use of what has been learned in one setting (Distribution P_1) to improve generalization in another setting (Distribution P_2). In transfer learning, the learner must perform two or more different tasks, but assume that many of the factors that explain the P_1 change are related to the changes that learning P_2 needs to grasp. In these cases, the goal is to take advantage of the data advantages of the first setting to extract information that may be useful in learning in the second setting or in making direct predictions. For example, identify a picture of a dog. These visual categories share some low-level concepts, such as edges, visual shapes, geometric changes, the effects of photo changes, etc. Therefore, transfer learning can be achieved by simply sharing low-level parameters to achieve better performance.

When the training model starts to update the parameters iteratively with the random initialization point as the starting point, because the signal-to-noise ratio is high, the samples can provide a better gradient to update the parameters, so the training is not very difficult, it is easy to get a good result. This is also because the number of layers is not very deep, the solution space is not very complex, resulting in any initial iteration can reach the global optimal point or close to it. However, as the SNR decreases, minibatch can only provide gradient estimation with severe noise disturbance, resulting in unstable training. In this case, only very small learning rates and carefully tuned hyper parameters can be used to train. But this greatly increased the workload of the experiment. Using transfer learning method and using the model parameters obtained under high signal-to-noise ratio to initialize low signal-to-noise ratio can greatly speed up the training speed and accuracy of the network.

Denoising Auto-Encoder. Denoising auto-encoder (DAE) is a kind of encoder which takes corrupted data as input and is trained to predict the original undamaged data. The DAE training process is shown in Fig. 5. First, a damage process $C(\tilde{x}|x)$ is introduced. This conditional distribution represents the probability of a given data sample x to produce a damaged sample $\tilde{x}$. The automatic encoder learns the reconfiguration distribution $p_{reconstruct}(x|\tilde{x})$ from the training data pair $(x, \tilde{x})$ according to the following procedure:

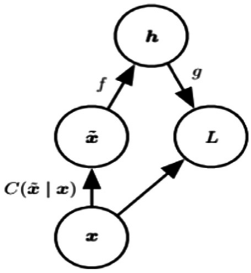

Fig. 5. Structure of denoising auto-encoder.

(1) Sampling a training sample from training data.
(2) Pick up a damaged sample $\tilde{x}$ from $C(\tilde{x}|x = x)$.
(3) $(x, \tilde{x})$ is used as a training sample to estimate the reconstruction distribution $p_{reconstruct}(x|\tilde{x}) = p_{decoder}(x|h)$ of the automatic encoder, where h is the output of the encoder $f(\tilde{x})$, and $p_{decoder}$ is defined according to the decoding function $g(h)$.

It is usually easy to minimize the gradient descent of the negative log likelihood $-\log p_{decoder}(x|h)$. As long as the encoder is deterministic, the denoising auto-encoder is a feedforward network and can be trained in the same way as other feedforward networks.

CNN is a suitable choice for the application of deep learning in signal demodulation. Convolution operation is a kind of operation that transforms time domain to frequency. Although the convolution form of convolution neural network is somewhat different, it is essentially similar. For signals, the characteristics are mainly reflected in the frequency domain. In addition, unlike most of the tasks performed by CNN, neural networks used in signals do not require a strong ability to extract features, which is attributed to the ease of feature extraction in signal demodulation. The use of high capacity deep networks can easily lead to overfitting. However, due to the poor robustness of the neural network to noise, the error performance of demodulation still needs to be improved. Experiments show that the difficulty of signal demodulation lies in how to resist noise rather than enhance the ability to extract features. Based on this analysis, this paper adopts the improved scheme of denoising auto-encoder pre-training and signal demodulation, and uses two-layer convolution neural network to build the encoding and decoder to de-noise the data. The final model training flow chart is shown in Fig. 6.

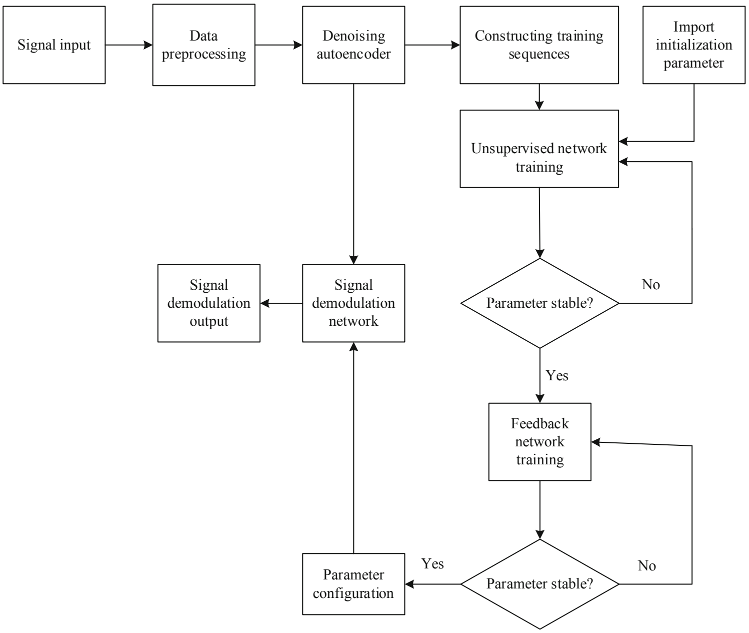
Signal input
Data preprocessing
Denoising autoencoder
Constructing training sequences
Import initialization parameter
Unsupervised network training
Parameter stable?
No
Yes
Feedback network training
Parameter stable?
No
Yes
Parameter configuration
Signal demodulation network
Signal demodulation output

Fig. 6. Flow chart of training mode.

4 Experiments and Results Analysis

In Fig. 7, the CNN model is used to demodulate a single modulated signal (FSK). Although there are some differences in BER performance compared with the coherent demodulation theory, the preconditions are different. The BER curve of the coherent demodulation theory is the precondition that the frequency interval equals the symbol transmission rate. The convolution neural network model is obtained under the premise that the frequency interval is equal to half of the symbol transmission rate. Therefore, the spectrum efficiency of this design is higher. At the same time, for incoherent demodulation, the frequency interval becomes smaller, which will greatly reduce the bit error performance. Under the same conditions, the demodulation performance of CNN model is 3.5 dB higher than that of incoherent demodulation. Compared with the best incoherent receiver, the performance of our model is better than that of the best incoherent detection when the signal-to-noise ratio is lower than 11 dB, and slightly lower than that of the best incoherent detection when the signal-to-noise ratio is higher than 11 dB. The reason for this analysis is that the test set data is not big enough. If we want to get the BER below 10–6 accurately, we must test at least one million data. Because of the limitation of hardware, we can’t detect such huge data at one time.

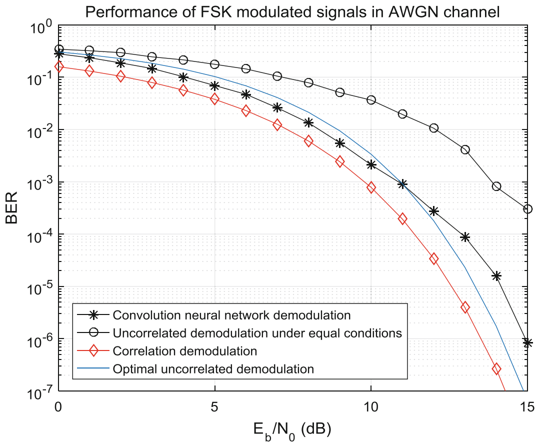

Fig. 7. Performance of FSK modulated signals in AWGN channel.

The demodulation performance curve of 4-PAM signal is shown in Fig. 8, and the improvement of the model by transfer learning and denoising auto-encoder is compared. The experimental results show that both transfer learning and denoising auto-encoder can improve the performance of the model by 0.5–1 dB, which indicates the demodulation performance of the network model is very close to correlated demodulation.

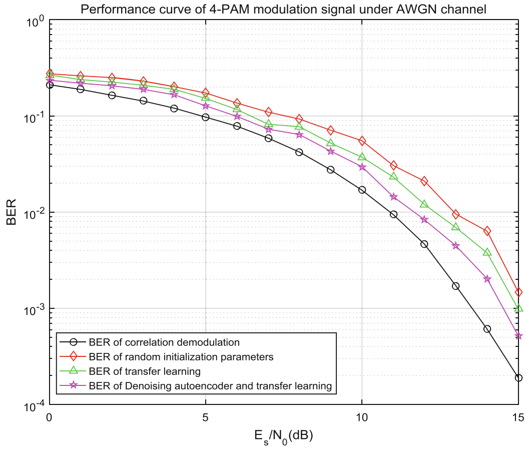

Fig. 8. Performance curve of 4-PAM modulation signal under AWGN channel.

Figure 9 shows the performance curve of the model for blind demodulation of mixed modulation signals. The experimental results show that the CNN model can blind demodulate the mixed modulation signals without knowing the modulation mode, and the improved model can improve the performance of blind demodulation about 2 dB. Through the comparative analysis of the experiments, we can see that the capacity and potential of the model are not fully realized when demodulating a single signal. The reason is that the characteristics of the signal needed for demodulation are not very large, so the performance advantages of the training methods of transfer learning and denoising auto-encoder are not obvious, but in a variety of ways. Under the condition of unified signal demodulation, the feature extracted by the model will increase linearly. At this time, the model is easy to converge to the local optimum, and the influence of noise will be greater. At this time, the performance of the model will be greatly improved. At the same time, after training the parameters, the network has good generalization performance, and it can achieve the expected goal of blind demodulation of multiple mixed modulation signals.

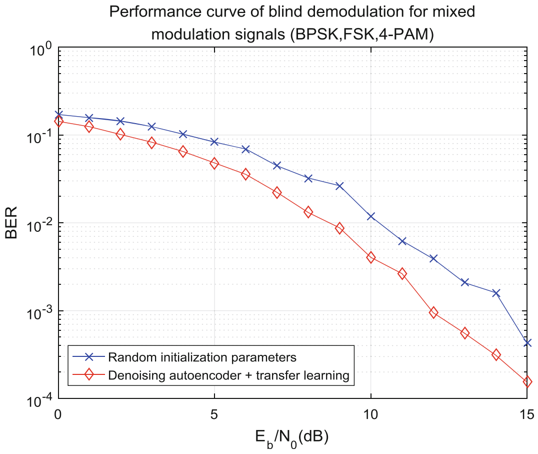

Fig. 9. Performance curve of blind demodulation for mixed modulation signals (BPSK, FSK, 4-PAM).

5 Conclusions

The main purpose of this paper is to design a blind demodulation method of mixed modulation signals based on convolution neural network. When transmitter transmits signals of different modulation modes at different time slots, the model can perform the corresponding demodulation tasks very well. At the same time, two methods of transfer learning and denoising auto-encoder are proposed for convolution neural network. The network model is improved and the performance of blind demodulation is improved by about 2 dB.

Acknowledgments. This work was supported by National Natural Science Foundation of China (Grant Nos. 61601147, 61571316) and Fundamental Research Funds of Shenzhen Innovation of Science and Technology Committee (Grant No. JCYJ20160331141634788).

References

1. Nielsen, M.A.: Neural Networks and Deep Learning. Determination Press (2015)
2. Simpson, A.J.R.: Abstract Learning via Demodulation in a Deep Neural Network. Computer Science (2015)
3. Milad, A.N., Aziz, M.M., Rahmadwati, A.: Neural network demodulator for quadrature amplitude modulation (QAM). Int. J. Adv. Stud. Comput. Sci. Eng. **5**(7), 10–13 (2016)
4. Elbaz, D., Zibulevsky, M.: End to End Deep Neural Network Frequency Demodulation of Speech Signals. arXiv preprint, pp. 1–6 (2017)
5. Mitra, U., Poor, H.V.: Neural network techniques for multi-user demodulation. In: IEEE International Conference on Neural Networks-Conference Proceedings, pp. 1538–1543. Institute of Electrical and Electronics Engineers Inc., San Francisco (1993)
6. Önder, M., Akan, A., Doğan, H., et al.: Advanced neural network receiver design to combat multiple channel impairments. Turkish J. Electr. Eng. Comput. Sci. **24**(4), 3066–3077 (2016)
7. Müller, J., Müller, J., Tetzlaff, R.: Hierarchical description and analysis of CNN algorithms. In: International Workshop on Cellular Nanoscale Networks and Their Applications, pp. 1–2. IEEE Computer Society, Notre Dame (2014)
8. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: 29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, pp. 4293–4302. IEEE Computer Society, Las Vegas (2016)