



A deep learning method based on convolutional neural network for automatic modulation classification of wireless signals

Yu Xu¹ · Dezhi Li¹ · Zhenyong Wang¹  · Qing Guo¹ · Wei Xiang¹

Published online: 5 March 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Automatic modulation classification plays an important role in many fields to identify the modulation type of wireless signals in order to recover signals by demodulation. In this paper, we contribute to explore the suitable architecture of deep learning method in the domain of communication signal recognition. Based on architecture analysis of the convolutional neural network, we used real signal data generated by instrument as dataset, and achieved compatible recognition accuracy of modulation classification compared with several representative structure. We state that the deeper network architecture is not suitable for the signal recognition due to its different characteristic. In addition, we also discuss the difficult of training algorithm in deep learning methods and employ the transfer learning method in order to reap the benefits, which stabilize the training process and lift the performance. Finally, we adopt the denoising autoencoder to preprocess the received data and provide the ability to resist finite perturbations of the input. It contributes to a higher recognition accuracy and it also provide a new idea to design the denoising modulation recognition model.

Keywords Modulation classification · Deep learning · Convolutional neural network · Wireless signal · Transfer learning · Denoising autoencoder

1 Introduction

Automatic modulation classification (AMC) is aiming to detect the modulation type of received signals in order to recover signals by demodulation. Research of modulation recognition techniques is one of the key technologies of receiver in the non-cooperative communication systems. It is of significance in both militarily and civilian applications. Recently, the idea of intelligent communication is

proposed and we hope the intelligent receiver can decode the message information and search for a specific signal for need. Furthermore, in military communication systems, detection of the modulation type is significant in generating energy efficient jamming signals.

Generally, the dominant approach of signal modulation recognition can be categorized as likelihood-based methods and feature-based methods [5]. Most likelihood-based classifiers require parameter estimation, while feature-based methods can be free from parameter estimation and achieve high popularity in recent years. Feature-based methods consist of two steps: feature extraction and classifier, which can provide classification decisions according to some particular criterion. Although there are many feature extracting methods and their combinations, however, their performance might not be easily compared as different algorithms consider different modulation sets and different assumptions, such as the channel fading, frequency carrier and phase offsets. However, most conventional feature-based methods cannot utilize full feature information when the performance of feature-based methods relies on the quality of the extracted features.

✉ Zhenyong Wang
ZYWang@hit.edu.cn

Yu Xu
xu_yu@hit.edu.cn

Dezhi Li
lidezhi@hit.edu.cn

Qing Guo
qguo@hit.edu.cn

Wei Xiang
wei.xiang@jcu.edu.au

¹ School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China

Furthermore, artificially choosing features is a complicated and difficult process. To solve this problem, some methods based on deep learning have been proposed and achieved good results [14, 20, 22], the advantage of which is that the method can extract features automatically from the signal data.

As a branch of machine learning, deep learning is a fascinating field and has achieved a series of state of the art results in different domains while it also has been tried for modulation classification in some related researches [14, 20, 22]. From the result show in the most literature, the deep learning method can get a better performance compared with the conventional feature-based method in most modulation sets. That is own to the information which is get from the large-scale training process. However, how to apply the deep learning method in the problem of communication signal recognition seems to not have a criterion.

In this paper, we contribute to explore the suitable architecture of deep learning method in the domain of communication signal recognition and pave the road for the development in automatic modulation classification. Among those algorithm architecture proposed in the literature, convolutional neural network (CNN) [12] enjoys high popularity due to its low complexity and translation invariance. We also adopt the CNN as our base structure to recognize the signal modulation types and get a good effect. Most research about deep learning method often highlight two salient aspects, which consist of how to design the model architecture and how to conduct training process. However, the result is varied sometimes it is even terrible due to the different training process. In our experiments we found that the transfer learning method [15], which refers to the situation where some learned knowledge is transferred to improve generalization in another setting, can help learning process. The motivation is that the same representation may be useful in both settings. The main contribution of transfer learning is stabilize the training process on account of a good initial point and get a lift in the performance. For improving performance further, we use the denoising autoencoder [19] to preprocess the received signal, which can bring prominent advantages to the next recognition management. In this case, we get the best performance in this paper in the last.

The main idea of this paper is to provide a stacked convolutional neural network of deep learning architecture for modulation classification based on extracted features of wireless signals automatically. We state the advantage of deep learning method and how we improve our model. Besides, we also analyze the improved performance method about the transfer learning and denoising autoencoder. The rest of the paper is organized as follows. In Sect. 2, convolutional neural network and some related

principle are investigated. Based on real sampled data of wireless signals, an improved CNN architecture is trained and proposed in Sect. 3. In Sect. 4, to solve the problem of training process, the transfer learning method is investigated and adopt to stabilize the training process. In Sect. 5, we introduce the principle of denoising autoencoder and how we training the denoising autoencoder architecture, which get a lift in the performance. Finally, conclusions are drawn in Sect. 6.

2 Convolutional neural network

2.1 Machine learning and deep learning

Machine learning has found wide-ranging applications in image/audio processing, finance and economics, social behavior analysis, project management, and so on. A machine learning algorithm is an algorithm that is able to learning from data. Machine learning algorithms can be simply categorized as supervised and unsupervised learning, where the adjectives “supervised/unsupervised” indicate whether there are labeled samples in the database. In this paper, we use supervised learning method to satisfy our demand. The goal of a machine learning model is to approximate a function f^* . For a classifier, function $y = f^*(x)$ maps an input x to a category y . A model defines a mapping criterion $y = f(x; \theta)$ and obtains the value of the parameters θ that result in the optimal approximation function of the true mapping function. Especially, deep learning allows the model to build complex concepts out of simpler concepts. The quintessential example of a deep learning model is the feed-forward deep network or multilayer perceptron (MLP). A multilayer perceptron is just a mathematical function mapping some set of input values to output values. The function is formed by composing many simpler functions.

2.2 Architecture of convolutional neural network

Convolutional neural network is a powerful architecture of artificial neural network, which is popular because of state of the art achievement in computer vision processing and natural language processing.

CNN process consists of two components: convolutional layers and pooling layers. Convolutional layers are comprised of filter kernels and feature maps. The filter kernels have weighted inputs and generate an output value like a neuron. The feature maps is the output of one filter kernel applied to the previous layer. A given filter kernel is drawn across the entire previous layer and moved one point at a time, which depends on the stride. Each position results in

activation of the neuron and generates an output to form the feature maps, as illustrated in Fig. 1.

Assume we have a 4D kernel tensor \mathbf{K} with element $K_{i,j,k,l}$ giving the connection strength between a unit in channel i of the output unit and a unit in channel j of the input, with an offset of k rows and l columns between the output unit and the input unit. Assume our input consists of observed data \mathbf{V} with element $V_{i,j,k}$ giving the value of the input unit within channel i at row j and column k . Assume our output consists of \mathbf{Z} with the the same format as \mathbf{V} . If \mathbf{Z} is produced by convolving \mathbf{K} across \mathbf{V} without flipping \mathbf{K} , then

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

where the summation over l, m and n is over all values for which the tensor indexing operations inside the summation is valid.

The pooling layer down-samples the feature map of previous layers. Pooling layers follow a sequence of convolutional layers to consolidate the learned features in the previous feature map. Therefore, pooling may be considered as a technique to compress and generalize feature representation, so as to generally reduce the model overfitting phenomena. In Fig. 2, the max pooling process is illustrated with pool width of 3 and stride of 2.

2.3 Training process based on stochastic gradient descent

Nearly all of deep learning is powered by important Stochastic gradient descent (SGD) algorithm [21]. Stochastic gradient descent is typical and preferred to training process for neural networks. One row of data is inputted into the network at a time. The network activates neurons forward to produce an output value finally. Then the output value is compared to the expected output value to generate an error value. The error is backward

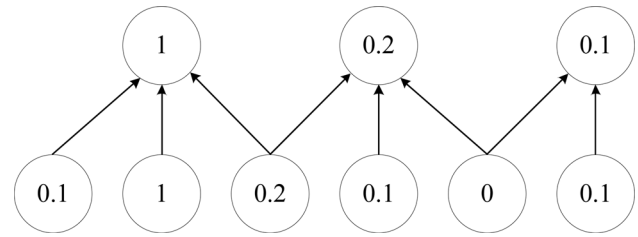


Fig. 2 Max pooling process of CNN

propagated through the network, in which the weights of layer are updated one after another, according to the contributed amount to the error. The process is repeated for all of the examples in the training data to get a trained network of the intended goal.

The weights in the network can be updated from the calculated errors for each training example, which can result in fast but also chaotic changes to the network. On the other hand, the errors can be saved up across all of the training examples and the network can be updated at the end.

The cost function used by a machine learning algorithm often decomposes as a sum over training examples of some per-example loss function. For example, the negative conditional log-likelihood of the training data can be written as

$$J(\theta) = E_{x,y \sim p_{data}} L(x, y, \theta) = \frac{1}{m} \sum_{i=1}^m L(x^{(i)}, y^{(i)}, \theta) \tag{2}$$

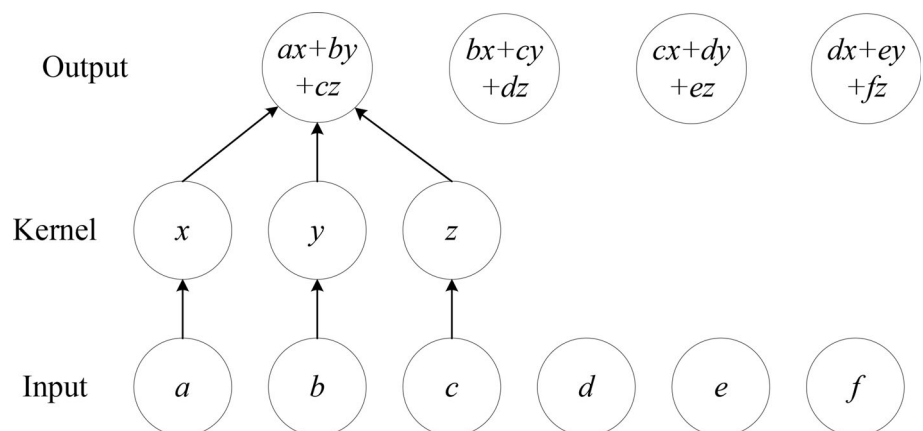
where L is the per-example loss

$$L(x, y, \theta) = -\log p(y|x; \theta) \tag{3}$$

For these additive cost functions, gradient descent requires computing

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

Fig. 1 Convolution operation of CNN



Considering the computational cost, we use a small set of samples to approximately estimate the true gradient. Specifically, on each step of the algorithm, we can sample a minibatch of examples $B = \{x^{(1)}, \dots, x^{(m)}\}$ drawn uniformly from the training set. The minibatch size m' is typically chosen to be a relatively small number of examples, ranging from 1 to a few hundred.

Based on examples from the mini-batch B , the estimate of the gradient is formed as

$$g = \frac{1}{m'} \nabla_{\theta} \sum_{i=1}^{m'} L(x^{(i)}, y^{(i)}, \theta) \quad (5)$$

The stochastic gradient descent follows the estimated gradient downhill

$$\theta \leftarrow \theta - \varepsilon g \quad (6)$$

where ε is the learning rate.

While stochastic gradient descent remains a very popular optimization strategy, learning with it can sometimes be slow. The method of momentum [16] is designed to accelerate learning, especially in the face of high curvature, small but consistent gradients, or noisy gradients. The momentum algorithm accumulates an exponentially decaying moving average of past gradients and continues to move in their direction. A hyperparameter $\alpha \in [0, 1]$ determines how quickly the contributions of previous gradients exponentially decay. The update rule is given by

$$v \leftarrow \alpha v - \varepsilon \nabla_{\theta} \frac{1}{m} \sum_{i=1}^m L(f(x^{(i)}; \theta), y^{(i)}) \quad (7)$$

$$\theta \leftarrow \theta + v \quad (8)$$

The velocity v accumulates the gradient elements $\nabla_{\theta} \frac{1}{m} \sum_{i=1}^m L(f(x^{(i)}; \theta), y^{(i)})$. The larger α is relative to ε , the more previous gradients affect the current direction.

3 The proposed CNN for modulation classification

To meet the requirements of modulation classification, our network architectures are mainly inspired by ALEXNET [11], as shown in Fig. 3.

3.1 Signals data sampled and process

Because digital modulation has better immunity performance to interference, which is mostly discussed in the literatures for modulation classification. Here, it is assumed that there is a single carrier-transmitted signal in additive

white Gaussian noise (AWGN) channel. The modulation types include 2ASK, BPSK, QPSK, 8PSK and 16QAM.

The signal data are produced by vector signal generator. The sampling rate is 1GHz. All the signal data of different modulation types have the same carrier frequency of 100 MHz and bandwidth of 25 MHz. Every sample has 2000 raw points and there are 25,000 samples in total, 5000 samples for each modulation type. The only preprocess is to rescale the amplitude to the range of -2 to 2 V.

For most classification and regression process, there is still possibility to get results even with small random noise added to the input. However, neural networks are proved not robust to noise [18]. One way to improve the robustness of neural networks is simply to do training process with input random noise data. So in training procedure to improve the robustness, training data of same SNR are included, which are also used to test the performance of proposed method in different SNR conditions.

When the network layers are not deep, it is not likely to encounter the problems like vanishing/exploding gradients [2, 4]. The principle of maximum likelihood is taken as the cost function, which means the cross-entropy between the training data and the prediction of the model is regarded as the cost function. The weights are initialized with Gaussian distribution initializers, which have zero means and unit variance. We use rectified linear units [13] as our activation function in every convolutional layers, the function is given by

$$g(z) = \max\{0, z\} \quad (9)$$

This makes the derivatives through a rectified linear unit remain large whenever the unit is active. We adopt the Softmax function as our output function which is given by

$$\text{softmax}(z)_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \quad (10)$$

The SGD is involved with a mini-batch size of 256. The weight decay is 0.0001 and the momentum is 0.9. The learning rates starts from 0.1.

As for the testing process, it is typically to use a simple separation of the same sampled data into training and testing datasets. In experiments, 80% data of the sampled signal is assigned to training dataset and 20% data of the sampled signal is assigned to testing dataset. Finally when the training is halt, we get the accuracy through inputting the testing datasets and statistical the accuracy.

3.2 The improved CNN architecture

In this section, we mainly explore how to design the adaptive model architecture to get a higher recognition accuracy and lower complexity.

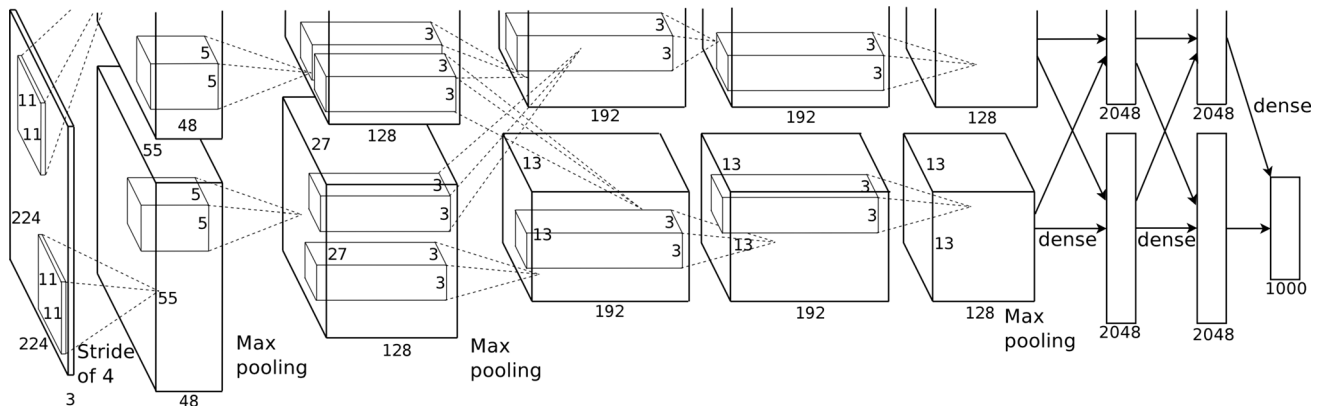


Fig. 3 The architecture of ALEXNET

To analyze the design principle and the result, we choose 3 kinds of representative structure proposed in the literatures and adopt the parameter to compare the performance and complexity. In our experiment, we use the stacked denoising sparse autoencoder proposed in [22] as the base structure to state that the advantage of CNN. Besides, paper [20] show that radio modulation recognition is not limited by network depth and we also investigate the deep neural network with more than 30 layers to explore the applicability. Finally, we also compare the performance with the CNN2 in paper [14] to demonstrate the role of the fully-connected layer.

In our improved CNN architecture, the large kernel size is designed for better performances and acceptable complexity. Moreover, after investigating the deep neural network with more than 30 layers, it is found that there are over-fitting problems. It is possible to apply a shallow neural network to compete modulation recognition for signals with reasonable SNR. Based on the analysis above, the number of input neuron is 2000, which means every sample has 2000 raw points. The improved CNN is proposed with 3 convolutional layers, and each convolutional layer is followed by a max pooling layer. At the end of the CNN network, a 5-way fully-connected layer with Softmax is used to output the probability of 5 kinds of signal modulations classification. The convolutional layers have filter kernels with length of 40. 64 filter kernels are used in both the input layer and the second layer. For the third layer, the filter kernels are increased to length of 128. The max pooling layers perform down-sampling with stride of 2 and pool width of 3 to get overlapping pooling. We do not use the any regularization like dropout [8]. So, the improved CNN consists of 4 weighted layers, as shown in Fig. 4.

3.3 Experiments and results analysis

In order to evaluate recognition performances of deep neural networks with shallow neural networks and explain what is overfitting phenomenon by an instance and explain how this phenomenon appear, a 32-layers RESNET [6, 7] (we have to use bottleneck structure due to the degradation problem) and the improved CNN with 4 layers are compared under condition of SNR = 0 dB.

As shown in Table 1, although both have similar training accuracy, the improved 4-layer CNN has better testing accuracy than the 32-layer RESNET. Overfitting occurs when the gap between the training error and test error is too large.

Although the deeper network can provide higher capacity and it will get worse result sometimes fail for two different reasons. First, the optimization algorithm used for training may not able to find the value of the parameters that corresponds to the desired function. Second, the training algorithm might choose the wrong function. Models with high capacity can overfit by memorizing properties of the training set that do not serve them will on the test set.

Recognition accuracy performances of the improved CNN are compared with several representative deep learning structure in different SNR conditions as shown in Fig. 5 and the complexity comparison is show in the Table 2.

The improved CNN has better recognition accuracy in these compared structure. Due to the overfitting problem, the performance of the RESNET structure is not good. But the descent speed in RESNET network is more stable than the improved CNN with shallow layers as the SNR drops to a low extent. The chain rule states that

$$\frac{dz}{dx} = \frac{dz}{dy} \frac{dy}{dx} \tag{11}$$

Fig. 4 The improved CNN structure

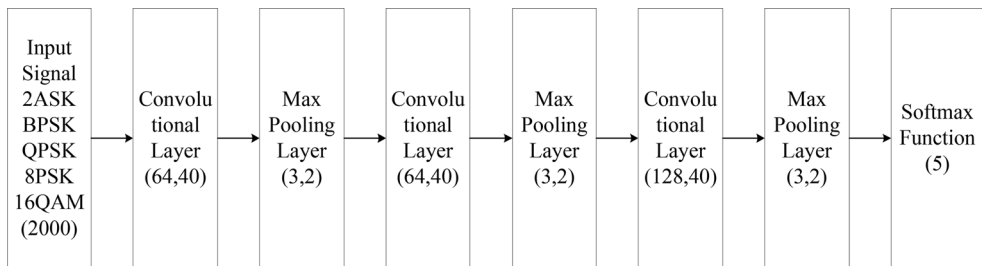


Table 1 The accuracy comparison between the improved CNN and RESNET (SNR = 0 dB)

	The improved CNN	32 Layers RESNET
Training accuracy	Approximate 100%	Approximate 100%
Testing accuracy	Approximate 100%	86.7%

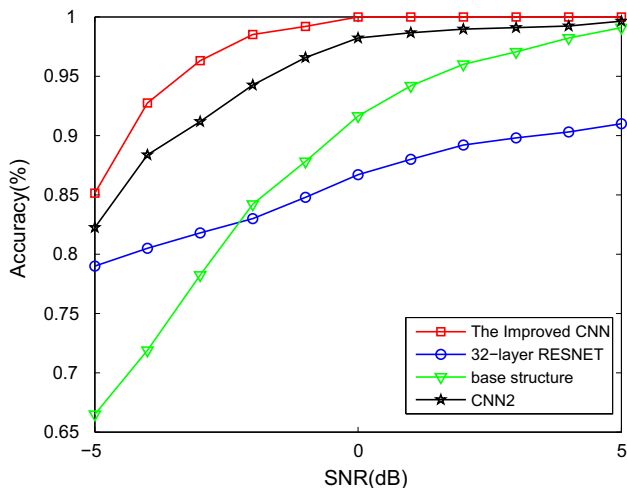


Fig. 5 Recognition accuracy comparison

Table 2 The complexity comparison (l is the number of input point)

	The number of multiplication
The improved CNN	$1.66 \times 10^5 \times l$
CNN2	$2 \times 10^5 \times l$
Base structure	$5 \times 10^3 \times l$

As the neural network becomes deeper, the derivatives of the expression of the neural network f is likely to be smaller, and the feature extraction function will be robust to finite perturbations of the input, which will be discussed detailed in Sect. 5. When the SNR of received signals is very weak, a deep neural network may provide stronger power to distinguish signal from noise.

Except the overfitting problem, all the CNN structure have a better performance than the base structure which is composed by multilayer perceptron. The features extracts by the CNN, which have the translation invariance, can utilize feature information better. And we found that the removal of the fully-connected layers of ALEXNET will reduce the amount of weight parameters and get little impact of the recognition accuracy performance. As a result we got our final structure and the improved CNN which has best recognition accuracy in these compared structure.

A prior to apply a deep neural network to a task is that the factors of variation which can explain the observed data are expressed in terms of other, simpler representation. And this simpler representation are combined to more complicated representation. That is why the deep learning method get so many success in some domains like computer vision. However, the communication signal is different from those conventional AI task so we think the deeper neural network is not appropriate for the signal recognition tasks.

4 Transfer learning improvement method

Transfer learning refers to the situation where what has been learned in one setting P_1 is exploited to improve generalization in another setting P_2 . In transfer learning, the learner must perform two or more different tasks, but we assume that many of the factors that explain the variations in P_1 are relevant to the variations that need to be captured for learning P_2 . A conventional example of transfer learning is show in Fig. 6. Sometimes what is shared among the different tasks is not the input but the output. It makes more sense to share the upper layers of the neural network. The transfer learning may help to learn representations that are useful to quickly generalize.

Using the same representation in both settings allows the representation to benefit from the training data that is available for both tasks. In this paper how we appple transfer learning method is show in Fig. 7. In this case, one task is easy to fulfill while another is hard due to the low quality

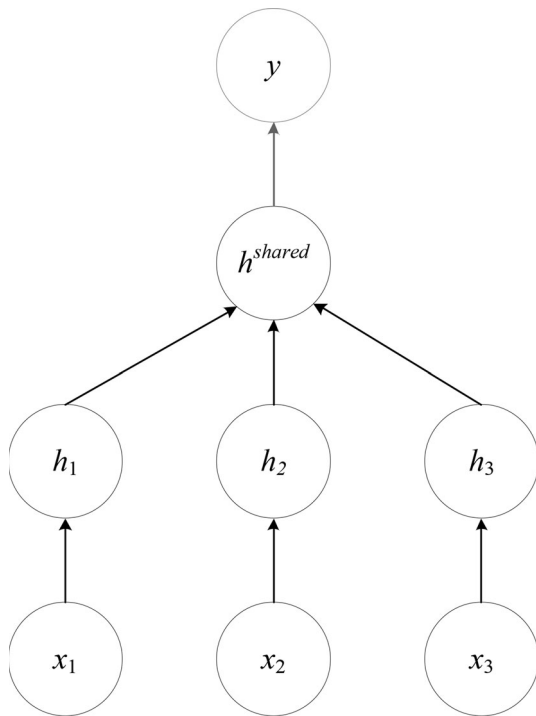


Fig. 6 An example of transfer learning sharing the output

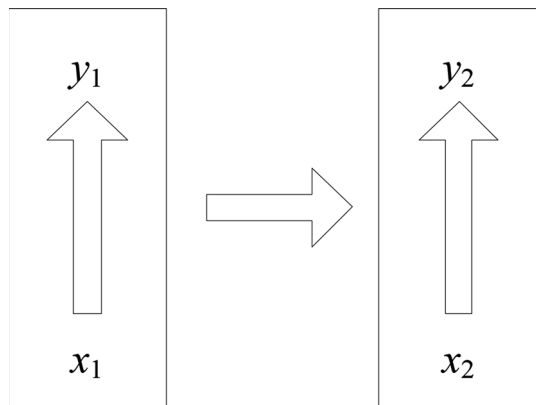


Fig. 7 Transfer learning of sharing weights

of dataset. We can use transfer learning through sharing weight between the two learning process.

We initialize the weights with the same initializer to explore the performance of the proposed neural network architecture in different SNR conditions and encounter the unstable training problem. We take the experiment with the SNR ranks -1 to -3 dB as an example to describe this

phenomenon. The result of three times in different SNR conditions is list in Table 3, we report “failed” when the accuracy is lower than 40%.

As can be seen in Table 3, when the SNR is high, the experiment works well and the proposed CNN gets satisfied recognize accuracy at approximate 100%. We contribute the reason to the dataset which can provide a good estimation of gradient to update the parameter so the initial point does not matter. An approximate global maximum point can be achieved. As the drop of the SNR, the model encounters the problem which we called unstable training. In this case, we think that the mini-batches give only a very noisy estimate of the gradient and it easy to fall into a local minimum point which blocks the training. The algorithm cannot calculate a right direction to move which contributes to a failure. The results fluctuated severely and a method based on transfer learning is proposed in the following.

We apply the weights trained in the high SNR conditions as the initial weights when we training the model in the low SNR conditions, which is a kind of transfer learning. We got the convergent point in high SNR conditions and the gradient become small. There are two tasks which have a same target to recognize the modulation type of the signal but have the different input. To some extent one task has a more clear input while the input of another task can be seen as a more polluted signal suffer from AWGN channel. Our experiment results show that the unstable training problem is well addressed in this setting. Besides, we also get a lift in the performance of accuracy.

We evaluate our method based on transfer learning by using the pre-training initial weights and the results is show in Fig. 8. The Line 1 is the performance of the original method using Gaussian distribution random initializers without pre-training. Due to the unstable training problem, we run the program 5 times and choose the best result as the final result. We first train the model on 0dB condition and it exists no difficult to get a satisfied recognize accuracy approximate 100%. We save the weights as the initial weights when we experiment the performance in other low SNR conditions and the result is show in Line 2. It is obvious that we obtain the accuracy gains from this method. Then we use the weights based on iterative SNR to explore the performance of the neural network in Line 3. For example, we initial the weights trained in the SNR of -1 dB when we train the model in the SNR of -2 dB and

Table 3 The accuracy comparison under various SNR conditions

SNR (db)	The 1st experiment	The 2nd experiment	The 3rd experiment (%)
-1	99.86%	99.44%	99.50
-2	Failed	97.31%	92.10
-3	92.08%	Failed	94.45

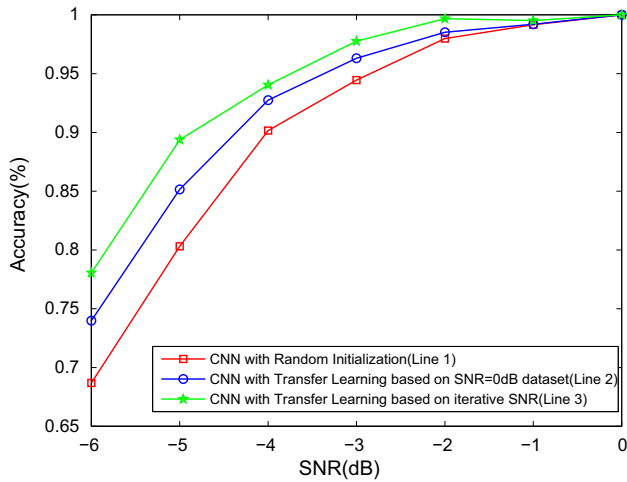


Fig. 8 Recognition accuracy comparison

recursively. Somehow surprisingly, the results are improved by healthy margins and report in Fig. 8. Here we assumed that there are several factors $A_1(t), A_2(t), \dots, A_n(t)$ that explain the input data mostly. As the SNR becomes lower, once the Eigenvalues of the factor is below the noise floor it will be hard to proceed the training process. However the factors is not vary as the SNR becomes lower. The initial point which shared in the high SNR conditions will make the training process close to the optimal point that stabilize the process and lift the performance.

5 Denoising autoencoder improvement method

An autoencoder [3, 9] is a neural network that is trained to attempt to copy its input to its output. Internally, it has a hidden layer h that describes a code used to represent the input. The network may be viewed as consisting of two parts: an encoder function $h = f(x)$ and a decoder that produces a reconstruction $r = g(h)$. This architecture is presented in Fig. 9.

The denoising autoencoder (DAE) is an autoencoder that receives a corrupted data point as input and is trained to predict the original, uncorrupted data point as its output. The DAE training procedure is illustrated in Fig. 10. We introduce a corruption process $C(\tilde{x}|x)$ which represents a conditional distribution over corrupted samples \tilde{x} . Given a data sample x , the autoencoder then learns a reconstruction distribution $p_{reconstruct}(x|\tilde{x})$ estimated from training pairs (x, \tilde{x}) .

Score matching [10] is an alternative to maximum likelihood. It provides a consistent estimator of probability distributions based on encouraging the model to have the

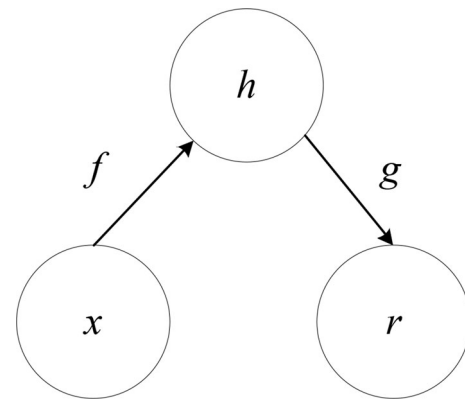


Fig. 9 The general structure of an autoencoder

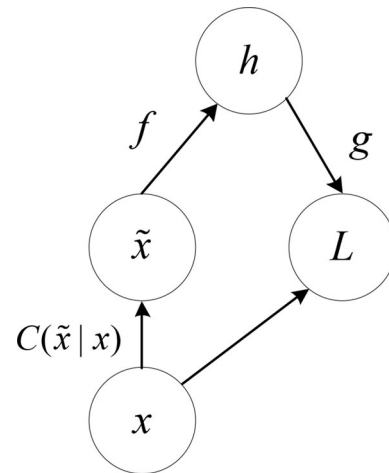


Fig. 10 The general structure of a denoising autoencoder

same score as the data distribution at every training point x . The training criterion of DAE makes the autoencoder learn a vector field $(g(f(x)) - x)$ that estimates the score of the data distribution. In this case, the score is a particular gradient field

$$\nabla_x \log p(x) \tag{12}$$

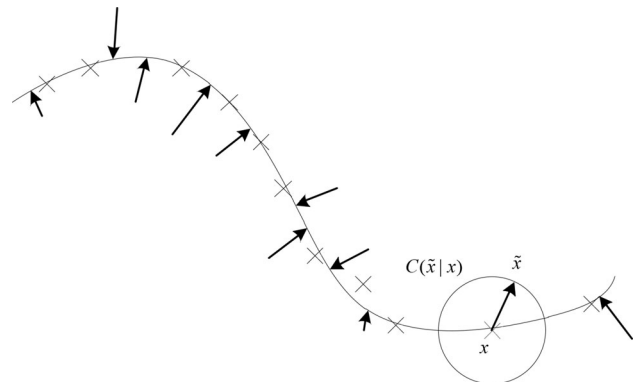


Fig. 11 The low-dimensional manifold of received signal and the training process

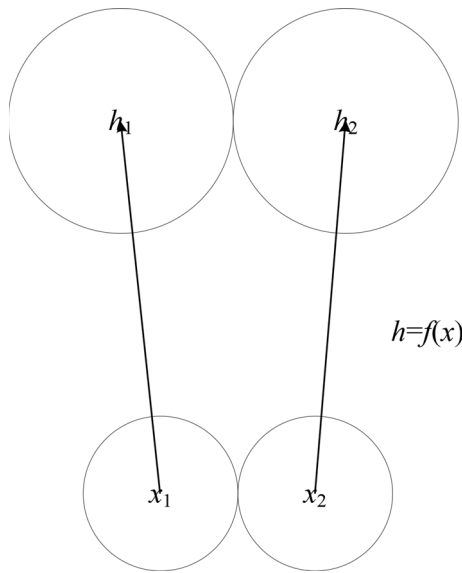


Fig. 12 A pedagogical explanation of denoising autoencoder

As illustrated in Fig. 11, a denoising autoencoder is trained to map a corrupted data point \tilde{x} back to the original data point x . We repress the training example x as crosses lying near a low-dimensional manifold. We illustrate the corruption process $C(\tilde{x}|x)$ with a circle of equiprobable corruptions. An arrow demonstrates how one training example is transformed into one sample from this corruption process. The vector $g(f(\tilde{x})) - \tilde{x}$ points approximately towards the nearest point on the manifold, since $g(f(\tilde{x}))$ estimates the center of mass of the clean points x which could have given rise to \tilde{x} .

In our paper we adopt score matching as our training criterion rather than maximum likelihood when we training

the denoising autoencoder. Besides, we also add an explicit regularizer

$$\Omega(h) = \lambda \left\| \frac{\partial f(x)}{\partial x} \right\| \tag{13}$$

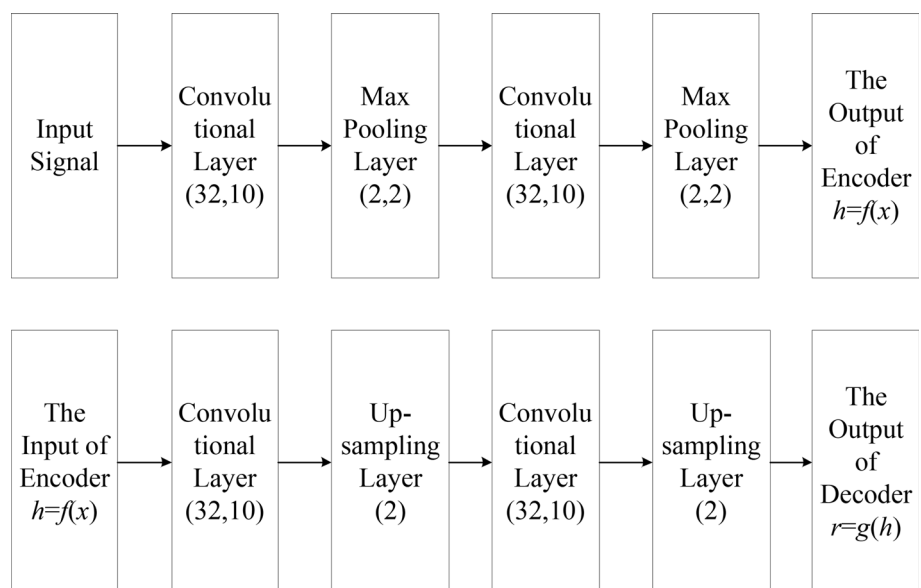
It is the core idea of the contractive autoencoder [17], which encourages the derivatives of f to be as small as possible. The penalty $\Omega(h)$ is the squared Frobenius norm of the Jacobian matrix of partial derivatives associated with the encoder function.

Paper [1] showed that in the limit of small Gaussian input noise, the denoising reconstruction error is equivalent to a contractive penalty on the reconstruction function that maps x to $r = g(f(x))$. Denoising autoencoders make the reconstruction function resist small but finite-sized perturbations of the input, while contractive autoencoders make the feature extraction function resist finite perturbations of the input.

As illustrated in Fig. 12, the encoder function transforms the input to another area, which is depend on the function $h = f(x)$. This kind of training criterion is encouraged to map a neighborhood of input points to a smaller neighborhood of output points. In other words, all perturbations of a training point x are mapped near to $f(x)$ and two different points x_1 and x_2 may be mapped to $f(x_1)$ and $f(x_2)$ points that are farther apart than the original points. This method is similar to increasing the code distance in coding principle.

In this Section, we use the convolutional neural network as our base structure of denoising autoencoder. The training detail is the same with the experiment in the Sect. 3 except that we choose the score matching with an explicit

Fig. 13 The structure of a denosing autoencoder



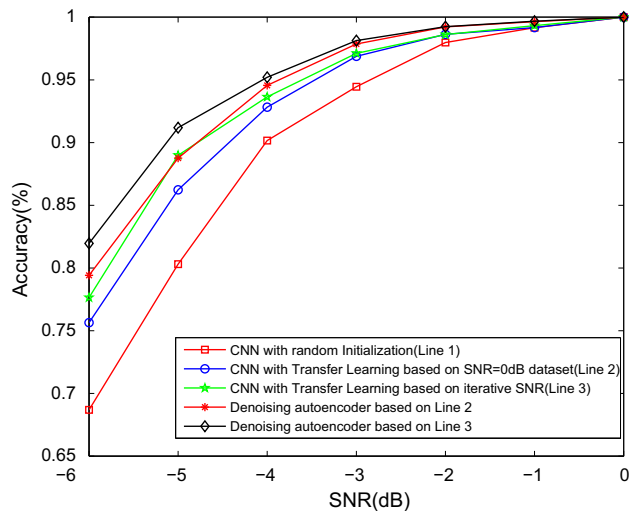


Fig. 14 Recognition accuracy comparison

regularizer as our training criterion. The structure is shown in the Fig. 13.

The final result curve is shown in the Fig. 14 where we compare the performance of 5 kinds of different methods. As can be seen from the figure, after processing by the denoising autoencoder, the input signal has a better representation than the former experiment. This method, that we combine the transfer learning method and denoising autoencoder method, improves the reorganization accuracy performance significantly.

6 Conclusion

The main idea of this paper is to explore the suitable architecture of deep learning methods in the domain of communication signal recognition and provide a stacked convolutional neural network based on extracted features of wireless signals automatically. There is not a criterion on how to combine the deep learning method and signal recognition nowadays. We also state that deep architecture is not suitable for this task in our experiment. It is different from the conventional AI tasks where the factors of variation which can explain the observed data are expressed in terms of simpler representation. As for the notorious problems in communication systems, noise interference, we think that transfer learning method is an effective method to stabilize the process. In order to further improve the performance, designing an appropriate structure using autoencoders may play an important role in the future.

Acknowledgements This work was supported by National Natural Science Foundation of China (Nos. 61601147, 61571316). Fundamental Research Funds of Shenzhen Innovation of Science and

Technology Committee (JCYJ20160331141634788), and the Fundamental Research Funds for the Central Universities (Grant No. HIT-MKSTISP. 2016013).

References

- Alain, G., & Bengio, Y. (2014). What regularized auto-encoders learn from the data generating distribution. *Journal of Machine Learning Research*, 15(1), 3563–3593.
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166.
- Bourlard, H., & Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4–5), 291.
- Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research*, 9, 249–256.
- Hazza, A., Shoaib, M., Alshebeili, S.A., & Fahad, A. (2013). An overview of feature-based methods for digital modulation classification. In: *1st International conference on communications, signal processing, and their applications (ICCSPA)* (pp. 1–6).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In *European conference on computer vision* (pp. 630–645).
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *Computer Science*, 3(4), 212–223.
- Hinton, G. E., & Zemel, R. S. (1993). Autoencoders, minimum description length and helmholtz free energy. In *International conference on neural information processing systems* (pp. 3–10).
- Hyvriinen, A., Hurri, J., & Hoyer, P. O. (2005). Estimation of non-normalized statistical models. *Journal of Machine Learning Research*, 6(4), 695–709.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(2), 2012.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 541–551.
- Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *International conference on international conference on machine learning* (pp. 807–814).
- O'Shea, T. J., Corgan, J., & Clancy, T. C. (2016). Convolutional radio modulation recognition networks. In *International conference on engineering applications of neural networks* (pp. 213–226).
- Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359.
- Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics & Mathematical Physics*, 4(5), 791–803.
- Rifai, S., Vincent, P., Müller, X., Glorot, X., & Bengio, Y. (2011). Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on machine learning, ICML 2011* (pp. 833–840).

18. Tang, Y., & Eliasmith, C. (2010). Deep networks for robust visual recognition. In *International conference on machine learning* (pp. 1055–1062).
19. Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. In *International conference on machine learning* (pp. 1096–1103).
20. West, N. E., & O’Shea, T. (2017). Deep architectures for modulation recognition. In: *IEEE international symposium on dynamic spectrum access networks (DySPAN)* (pp. 1–6).
21. Yazan, E., & Talu, M. F. (2017). Comparison of the stochastic gradient descent based optimization techniques. In *International artificial intelligence and data processing symposium (IDAP)* (pp. 1–5).
22. Zhu, X., & Fujii, T. (2017). A modulation classification method in cognitive radios system using stacked denoising sparse autoencoder. In *IEEE Radio and Wireless Symposium (RWS)* (pp. 218–220).



Yu Xu was born in 1995. He received his B.S. degree in Information and Communication Engineering from Harbin Institute of Technology (HIT) in 2017. He is currently pursuing the Ph.D. degree with Information and Communication Engineering, Harbin Institute of Technology, Heilongjiang, China. His current research interests include machine learning in communication system.



Dezhi Li was born in 1981. He received his M.E. and Ph.D. degree in Information and Communication Engineering from Harbin Institute of Technology (HIT) in 2006 and 20012 respectively. He is now with the Harbin Institute of Technology as a Research Assistant and Master Supervisor. He chaired a National Natural Science Foundation Youth Science Foundation and participated in the National Natural Science Foundation of China, the National

863, national defense basic research and other major national projects. His current interests include satellite communication, digital signal processing, fountain codes and software radio.



Zhenyong Wang was born in 1977. He received his M.E. and Ph.D. degree in Information and Communication Engineering from Harbin Institute of Technology (HIT) in 2002 and 2007 respectively. He worked for Department of Electronics and Communication Engineering HIT as an Associate professor since 2012. During April 2011 and April 2012, he was at the University of Sydney in Australia as a visiting scholar. He is now with the Harbin Institute of

Technology as an associate professor, doctoral supervisor and IEEE senior member. He had accomplished important projects of National Natural Science Foundation of China, international cooperation of Chinese Ministry of Education, the National Basic Research Program of China, and National High Technology Research and Development Program of China. He has published over 30 research papers, book of <Satellite Communication Systems> and nine authorized patents. His current interests include Satellite TT&C, Space Information Networks, Wireless Cognitive Radio, Internet of Things and Artificial Intelligence.



Qing Guo was born in 1964. He received his M.E and PH.D degree in Information & Communication Engineering in 1990 and 1998 respectively, all from Harbin Institute of Technology (HIT), Harbin, China. He is currently the Dean, professor and doctoral supervisor of the school of electronics and information engineering in Harbin Institute of Technology. He is the director of the Key Laboratory of broadband wireless communication and network in

Heilongjiang province and the deputy director of the special communication system engineering research center of the Ministry of education. His current interests include satellite communication, deep space communication, space information network and scientific research on wireless transmission and network.



Wei Xiang received the B.E. and M.E. degrees, both in electronic engineering, from the University of Electronic Science and Technology of China, Chengdu, China, in 1997 and 2000, respectively, and the Ph.D. degree in telecommunications engineering from the University of South Australia, Adelaide, Australia, in 2004. He is currently the Founding Professor and Head of Discipline of Internet of Things Engineering at James Cook University,

Cairns, Australia. During 2004 and 2015, he was with the School of Mechanical and Electrical Engineering, University of Southern Queensland, Toowoomba, Australia. He is an elected Fellow of the

IET and Engineers Australia. He received the 2016 TNQ Innovation Award, and 2017 Pearcey Entrepreneurship Award (Highly Commended), and 2017 Cairns Engineer of the Year. He was a co-recipient of three Best Paper Awards at 2015 WCSP, 2011 IEEE WCNC, and 2009 ICWMC. He has been awarded several prestigious fellowship titles. He was named a Queensland International Fellow (2010–2011) by the Queensland Government of Australia, an Endeavour Research Fellow (2012–2013) by the Commonwealth Government of Australia, a Smart Futures Fellow (2012–2015) by the Queensland Government of Australia, and a JSPS Invitational Fellow jointly by the Australian Academy of Science and Japanese Society

for Promotion of Science (2014–2015). He is the Vice Chair of the IEEE Northern Australia Section. He was an Editor for IEEE Communications Letters (2015–2017), and is an Associate Editor for Springer's Telecommunications Systems. He has published over 200 peer-reviewed journal and conference papers. He has severed in a large number of international conferences in the capacity of General Co-Chair, TPC Co-Chair, Symposium Chair, etc. His research interest falls under the broad areas of communications and information theory, particularly the Internet of Things, and coding and signal processing for multimedia communications systems.