Communication Radiation Source Identification Based on Semi-supervised Learning

Congxia Wu[1] 23020231154233,
Lamu QiWang [1] 23320230157384, Zige Tian[1] 23020231154226,
[1]Information Class

Abstract

Specific emitter identification (SEI) technique is an important research topic in the field of communication countermeasures. In recent years, the mainstream research direction in the field of SEI has gradually changed from the traditional method to the SEI method based on deep learning which has stronger feature extraction and expression ability. At present, deep learning-based research on SEI mainly relies on a lot of labeled samples and uses supervised learning to realize individual identification of various communication radiation sources. However, in the actual environment, we can obtain very few labeled samples, if only supervised methods are used for classification and recognition, the classification effect is not good. Therefore, a semi-supervised SEI algorithms are proposed to solve the problem of few-labeled samples. The main work of our team is as follows: A semi-supervised communication transmitter individual identification algorithm based on Stacked Denoising Autoencoders (SDAE) is proposed. Using the multi-level abstract feature learning ability of Stacked Autoencoders (SAE) and noise reduction capabilities of Denoising Autoencoder (DAE), the algorithm can effectively classify the different radiation sources under the condition of few-labeled samples.

Introduction

With the continuous development of mobile communication technology and the continuous popularization of wireless network equipment, the demand for identifying different communication radiation source equipment is increasing day by day. Specific emitter identification (SEI) technique has gradually become a research hot-spot in the field of wireless communications.

The process of SEI mainly includes four steps: signal reception, signal processing, feature extraction and classification and identification.

First, after the signal receiving system receives the signal emitted by the communication radiation source, it performs various processing on the received signal such as filtering, denoising, and power normalization; After that, the signal samples are transformed into feature domains based on the modulation parameters, high-order features and other information of the signal; Then, feature extraction is performed on the signal; Finally, through the classification and recognition of features, the SEI is achieved.

SEI can be divided into SEI based on shallow learning and SEI based on deep learning. Traditional SEI technique is based on shallow learning, which manually extracts features and imports them into a classification model for individual identification. For example, support vector machine, autoencoder, etc. However, it has some obvious shortcomings, such as: manual extraction requires the support of expert experience, poor generalization, the process is cumbersome and susceptible to interference from complex and changeable environments, etc. It can be known from the above shortcomings that in the face of complex and changeable electromagnetic environments, traditional shallow methods have been difficult to meet the needs of various complex application scenarios.

SEI based on deep learning uses a deep neural network to automatically extract the intrinsic features of the signal to replace the traditional manual feature extraction, thus avoiding the complicated feature extraction process used in traditional methods. Deep neural networks are different from shallow models in traditional methods. They have more layers of deep models, so they have stronger feature learning and expression capabilities. Deep learning methods can more effectively learn and understand complex features in data, thereby improving recognition accuracy and efficiency. It can also reduce manual intervention and errors, helping us better cope with high-dimensional and complex data processing and analysis work. SEI based on deep learning has many advantages over traditional methods. It is gradually replacing traditional machine learning methods based on manual feature extraction and becoming a new research hotspot.

Related works

Research on SEI based on deep learning in recent years generally assumes that a large amount of communication radiation source observation data with category labels is obtained, and the classification of individual communication radiation sources is mainly completed through supervised learning. However, in an actual environment, we can obtain a large number of unlabeled communication radiation source individual samples but can only obtain a small number of labeled communication radiation source observation samples. In this case, it is obviously impossible to effectively realize SEI by directly using supervised feature extraction and classification identification methods.

Semi-supervised learning is an important branch of machine learning, aiming to improve model performance by leveraging datasets that have both labeled and unlabeled samples. What sets semi-supervised learning apart from supervised learning is its effective utilization of information from unlabeled data, thereby enhancing the model's generalization capability.

In the research of semi-supervised learning, researchers have proposed many effective methods to fully utilize unlabeled data. Among them, graph-based approaches construct a graph structure based on the similarity between data points and utilize this structure for semi-supervised learning. Additionally, generative models have been widely applied to semi-supervised learning tasks, achieving the goal by modeling the data generation process. Furthermore, in recent years, significant progress has been made in the development of deep semi-supervised learning methods, which utilize deep neural network architectures for feature learning and representation, thereby improving the performance of semi-supervised learning tasks.

Moreover, semi-supervised learning has achieved significant results in various fields, including image recognition, natural language processing, and social network analysis. For example, in the field of image processing, semi-supervised learning methods that combine labeled and unlabeled images can significantly enhance the performance of image classification and segmentation. In natural language processing, semi-supervised learning has been widely applied to tasks such as sentiment analysis and named entity recognition, achieving notable effects. Semi-supervised learning provides an effective solution for the challenge of acquiring labeled data in practical applications, offering an important means to enhance model generalization capability and performance.

Approach

1 Denoising Autoencoder

The idea of Denoising Autoencoder Network was first proposed by Vincent et al. [1] in 2008. They learn high-order features of the input data by adding noise to the input data and using autoencoders for noise reduction. This method can not only effectively remove noise in the input data, but also improve the robustness and generalization ability of the model. Autoencoder Network is an unsupervised learning model, which contains two key parts: encoder and decoder. The encoder converts high-dimensional input samples into low-dimensional latent variables so that the neural network can learn the most meaningful features; the decoder restores the latent variables of the hidden layer to the original dimensions. During the training process, when the reconstructed sample output by the encoder in the Autoencoder Network can be as consistent as the input sample, it means that the encoder can extract important features that can restore the original sample, and it also means that the training effect of the model is relatively good. The basic structure of the autoencoder is shown in Figure 1.

The principle of the Denoising Autoencoder Network is to contaminate the input samples with artificial noise on the basis of the Autoencoder Network, so that some of the learnable features are lost, and then put them into the Autoencoder Network for learning, and require the reconstruction of the complete input samples. Adding noise to the sample will make the entire feature extraction and encoding process more robust, which means that the features extracted by the Autoencoding Network will have better generalization capabilities and better anti-noise capabilities. The basic principle of the Denoising Autoencoder Network is shown in Figure 2.

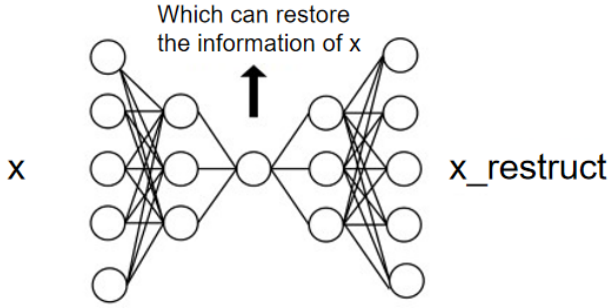

Figure 1: Structure diagram of Autoencoder Network

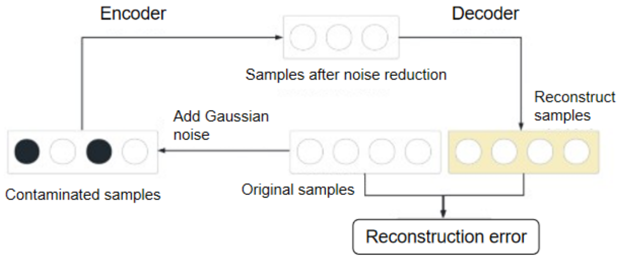

Figure 2: Schematic diagram of Denoising Autoencoder Network

2 Stacked Autoencoders

Stacked Autoencoder Network (SAE) was first proposed by Geoffrey Hinton et al. [2] in 2006. Stacked Autoencoder Network is a classic deep learning method. It is a deep neural network that is stacked by multiple autoencoders and has a larger number of layers and a more complex network. Its simple model structure is shown in Figure 3.

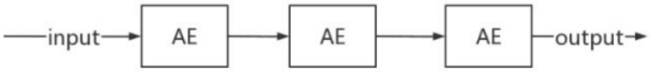

Figure 3: Structure of Simple Stacked Autoencoder Network

Compared with general autoencoder networks, Stacked Autoencoder Networks can extract high-order features of data layer by layer by stacking multiple autoencoders. Each layer extracts useful features from the output of the previous layer, which enables the network to better discover the combined features of the data. In more complex radiation source identification tasks, using SAE can improve the accuracy of classification and identification through deep feature extraction and make the model more robust.

3 Stacked Denoising Autoencoder

In this experiment, we stacked multiple denoising autoencoders and designed a stacked denoising autoencoder network. This network combines SAE's deep ability to extract subtle features of signals and DAE's good robustness to construct a deep network with higher accuracy and more robust recognition capabilities.

3.1 frame The first part is to train an unsupervised stacked denoising autoencoder network. First, the unlabeled radiation source samples are denoised, and then input into the stack autoencoder network for training. The restored samples output by the decoder are obtained, and the reconstruction error term is constructed together with the input samples to form the cost function required for training. Then, the back propagation algorithm is used to optimize the weight coefficients of each layer of the entire network, and the denoising autoencoding network is continuously trained so that the input and output sample distributions become consistent.

The second part is supervised classifier training. After unsupervised training of the denoising autoencoder network, the labels of the labeled communication radiation source samples are extracted, and the unlabeled samples and extracted labels are saved as $X2_{in}$ and $X2_{label}$ respectively. Then, input the extracted label $X2_{in}$ into the denoising autoencoder network for feature extraction and return the feature $a^{(L)}$ output by the encoder. After that, the extracted feature $a^{(L)}$ and the corresponding label $X2_{label}$ are used to construct the cross entropy loss and optimize the weight coefficient of the classifier through the back propagation algorithm to achieve the purpose of training the supervised training classifier. Finally, input the test sample for individual identification and analyze the experimental results.

3.2 Model training The input unlabeled communication radiation source observation sample is $X_{in} = [x_1...x_n]$, first add random Gaussian noise to each sample, then the sample after adding the noise is $\widehat{X} = [\widehat{x}_1...\widehat{x}_n]$:

$$\widehat{x} = x + \epsilon, \epsilon \in N(0, 0.01) \tag{1}$$

Among them, ϵ represents random Gaussian noise with a mean of 0 and a variance of 0.01.

The sample input to the autoencoding network is $\widehat{X} = [\widehat{x}_1...\widehat{x}_n]$, then the encoding function of the autoencoding for each input sample x is:

$$h(\widehat{x}) = \omega\widehat{x} + b \tag{2}$$

Among them, ω is the weight coefficient of the hidden layer neuron, and b is the hidden layer bias. For the l-th layer of the hidden layer, the nonlinear excitation of the neuron output is:

$$a^{(l)} = f(h^{(l-1)}) = LeakyReLU(h^{(l-1)}) \tag{3}$$

The training of the decoder is to first use the output of the stacked encoder as input, and then stack the outputs of the two decoders to reconstruct y. It is known that the 2L layer is the output layer, and the output reconstruction y is:

$$y = a^{(2L)} = f(h^{(2L-1)}) \tag{4}$$

Define the mean square error between the reconstruction y and the high-dimensional input $X1_{in}$ as the reconstruction error, and then construct the cost function $J(w, b)$. In order to prevent overfitting, the L2 regular term was also added to the experiment:

$$J(w, b) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\|x - y\|_2{}^2 + \lambda\sum_{i=1}^{k}\omega_i^2 \tag{5}$$

Among them, the first term is the reconstruction error term in the unsupervised training network, where N refers to the number of input samples; the second term is the L2 regular term, where is the λ weight attenuation factor.

After obtaining the cost function, the cost function is iteratively optimized through the back propagation algorithm to find its minimum value and continuously optimize the network.

After training the denoising autoencoder network with unlabeled communication radiation source samples, the labeled data set is used to train the linear classifier. The main method is to input the de-labeled $X2_{in}$ into the denoising autoencoder network for feature extraction and return the hidden quantity $a^{(L)}$ output by the encoder, and pass $a^{(L)}$ through the linear classification layer to obtain the classified data $Z = [z_1...z_n]$, then put it into the cross-entropy loss function together with the corresponding label $X2_{label} = [x_{l_1}...x_{l_n}]$ to calculate the $loss$. Here z will first convert the data into a vector $P = [p_1...p_n]$ representing the probability distribution through softmax:

$$p_i = P(z_i) = \frac{e^{z_i}}{\sum_i e^{z_i}} \tag{6}$$

Among them, z_i represents the data after each classification, and $P(z_i)$ is the calculated probability distribution of each category of each data.

After obtaining the probability distribution $P = [p_1...p_n]$, you can calculate the cross entropy loss:

$$loss = -\sum_{i=1}^{n} x_{l_i} log(p_i) \tag{7}$$

Among them, p_i represents the probability distribution of each category of each data, x_{l_i} represents the corresponding label vector (one-hot type), and n represents the number of data.

After obtaining the $loss$, the cost function is iteratively optimized through the back propagation algorithm to find the minimum loss value and continuously optimize the network to achieve the purpose of supervised training of the encoder.

3.3 Network design In this experiment, for the unsupervised training part, an autoencoder based on convolutional neural network extraction features was designed. Conv1d and ConvTranspose1d were used in the design of the encoder and decoder, and set their $kernel\_size = 2$, $stride = 2$, $padding = 0$. The entire encoder uses 4 layers of Conv1d (using LeakyReLU as the activation function) and 4 BatchNorm1d layers alternately stacked, as well as the last regularization layer (Dropout layer). The entire decoder uses 4 BatchNorm1d layers and 4 layers of ConvTranspose1d (using LeakyReLU as the activation function) alternately stacked and the last regularization layer (Dropout layer). It is worth mentioning that since one of the key points of unsupervised learning is to prevent overfitting, when designing the autoencoder, a Dropout layer and a BatchNorm1d layer were added to further reduce overfitting.

The design of the supervised training part is mainly to classify the low-dimensional features output by the encoder that can restore the input sample through a linear classifier, and then put them into the cross-entropy loss function (CrossEntropy) together with the corresponding label $X2_{label}$ to calculate the loss. Finally, the back propagation algorithm is used to optimize the classifier to achieve supervised training of the classifier.

3.4 Experimental steps Step 1: Input the unlabeled radiation source sample set $X1_{in}$ and put it into the DAE network for unsupervised feature extraction. According to Equation (1), noise is added to the input data to obtain the contaminated sample set $\widehat{X} = [\widehat{x}_1...\widehat{x}_n]$, which is then input into the network according to Equation (2).

Step 2: Construct an unsupervised training cost function. Calculate the reconstructed y through the decoder, and then construct the cost function J according to Equation (4).

Step 3: Iteratively calculate the unsupervised cost function through the backpropagation algorithm to continuously optimize the network.

Step 4: After training the DAE with the unlabeled sample set, input the unlabeled $X2_{in}$ into the DAE for feature extraction and return the hidden quantity $a^{(L)}$ output by the encoder. Then, Z is obtained after passing $a^{(L)}$ through the linear classification layer. Finally, put Z and the corresponding label $X2_{label}$ together into the cross-entropy loss function to calculate its cost function according to Formula (5), Formula (6), and Formula (7).

Step 5: Optimize the weight coefficient of the classifier through the back propagation algorithm to achieve the purpose of training a supervised training classifier.

Step 6: Use the test set for individual identification and output the classification results.

Experiment

In this section, we evaluate the performance of the SDAE network and compare it with the performance of unsupervised algorithms. In our experiments, the SEI network architecture is built on Python 3.9 in Pytorch, and the network is trained on NVIDIA GeForce GTX 1050.

1 Dataset The real-world signals used in the experiment were collected as IQ data for a total of 8 categories, with 1000 samples each and 1024 data points per sample.Additionally, the signal is passed through various signal-to-noise ratios (SNRs) and channel situations include AWGN, Rayleigh fading, and Rice fading channels.

In this experiment, we need to conduct different control experiments based on the proportion of labeled samples. Therefore, we divide each subsample into three sample sets: labeled training sample set, unlabeled training sample set and test sample set. After the data set is divided, different experimental conditions are divided according to the different proportions of the number of labeled training samples in the total samples, which are E1, E2, E3 and E4 respectively. Under condition E1, the number of labeled samples accounts for 10% of the total number of samples; under condition E2, the number of labeled samples accounts for 15% of the total number of samples; under condition E3, the number of labeled samples accounts for 20% of the total number of samples. The number of unlabeled samples is fixed at 60% of the total number of samples, and the number of test samples is 10% randomly selected from the remaining 20% of samples. Under the E4 condition, unsupervised training is performed, that is, the number of labeled samples is 0, and the number of unlabeled samples is fixed at 60% of the total number of samples.

2 Configuration Details

In the experiments of this chapter, the specific parameter settings for model training are as follows: set batch_size1 of the training set to 40, and set batch_size2 of the validation set to 20. The batch_size2 of the test set is set to 20. The batch times (steps_per_epoch) of the training set, test set, and validation set are set to 140, 20, and 40 respectively. epochs is set to 100 times. The dropout ratio in the encoder regularization layer is 0.3 and the dropout ratio in the decoder regularization layer is 0.9. The optimizer selected for unsupervised training of the network is ASGD, and the learning rate is 0.001. The optimizer when setting the supervised training classifier is Adam, and the learning rate is 0.006.

Under the four divided experimental conditions, the algorithm designed in this chapter is used to conduct experiments on the data set. Each experiment was repeated 30 times, and the average recognition rate of 30 times was taken as the evaluation index.

3 Experimental results and analysis

The average recognition rate and its standard deviation of the SDAE method on the USRP dataset are shown in Table 1.

According to the results in Table 1, it can be seen that using the semi-supervised communication radiation source algorithm based on SDAE can extract the characteristics of a large number of unlabeled samples and classify and identify them, effectively dealing with the unfavorable situation of a small number of labeled samples in actual situations. . Compared with unsupervised radiation source identification, this algorithm can accurately classify and identify with fewer labeled samples. In addition, we can see that as the number of

Table 1: Average recognition rate of SAE method on USRP dataset

Experiment environment	Training set recognition accuracy	Test set recognition accuracy
E3	88.67 (±3.14)	78.98 (±3.43)
E2	84.61 (±4.97)	78.24 (±4.31)
E1	82.46 (±6.32)	77.51 (±5.37)
E4	26.67 (±2.21)	22.45 (±1.63)

labeled samples increases, the average recognition rate will not increase significantly, which shows that the algorithm model has a certain degree of robustness in addition to good robustness.

Conclusion

This paper studies the problem of less available information in the process of identifying communication radiation sources in actual situations, based on the advantage that semi-supervised learning can use a small amount of labeled data and a large amount of unlabeled data to train the network, a semi-supervised communication radiation source individual identification algorithm based on stack denoising autoencoding network is proposed. This algorithm cleverly combines the advantages of stacked autoencoder networks and denoising autoencoder networks, showing excellent robustness and accurate feature recognition capabilities. In the case of small samples, it is possible to accurately extract signal fingerprint features and realize radiation source classification and identification.

This article is mainly aimed at individual identification of communication radiation sources when there are a small number of tags, but the actual situation may be worse than the conditions set up in the experiment. Therefore, in the future, research can be carried out in the direction of small sample problems with more stringent research conditions. Regarding the small sample problem, research can continue from the following aspects in the future:

1. The crux of the small sample problem lies in the small number of data samples. In order to solve this problem, data augmentation method can be adopted, that is, using existing data sets to generate false samples similar to real samples to expand the data set. Doing so can improve the accuracy of classification recognition.

2. Can study metric learning methods that can solve small sample problems. Metric learning refers to the technology of learning the distance between samples. This method measures the similarity between samples to minimize the distance between similar samples and maximize the distance between samples of different categories.

3. Multi-task learning models can be used to solve small sample problems. Multi-task learning models mainly exploit the correlation between different tasks to share underlying features. Then, unique features of tasks are learned for different tasks, and the number of parameters that need to be learned for new tasks is effectively reduced through parameter sharing.

References

Dai, Z.; Yang, Z.; Yang, F.; Cohen, W. W.; and Salakhutdinov, R. R. 2017. Good semi-supervised learning that requires a bad gan. *Advances in neural information processing systems*, 30.

Hinton, G. E.; and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *science*, 313(5786): 504–507.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kwok, H. K.; and Jones, D. L. 2000. Improved instantaneous frequency estimation using an adaptive short-time Fourier transform. *IEEE transactions on signal processing*, 48(10): 2964–2972.

Learning, S.-S. 2006. Semi-Supervised Learning. *CSZ2006. html*.

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2536–2544.

Rasmus, A.; Berglund, M.; Honkala, M.; Valpola, H.; and Raiko, T. 2015. Semi-supervised learning with ladder networks. *Advances in neural information processing systems*, 28.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *nature*, 323(6088): 533–536.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, 1096–1103.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 912–919.