Spectrum Data Reconstruction via Deep Convolutional Neural Network

Xiaojin Ding(✉) and Lijie Feng

Telecommunication and Network" National Engineering Research Center, Nanjing University of Posts and Communications, Nanjing 210003, China
dxj@njupt.edu.cn

Abstract. In the paper, we explore the spectrum-data reconstruction of a spectrum-sensing system. In order to decease the demand on the sensed spectrum data, we proposed a deep convolutional neural network (DCNN) based spectrum data reconstruction scheme relying on three stages, thus the satellites are allowed to perform spectrum sensing with the aid of down-sampling, and transmit the low-resolution (LR) and small amount of high-resolution (HR) spectrum data to earth stations. Specifically, in the first stage, the received LR and HR spectrum data will be first preprocessed. Then, the preprocessed HR spectrum data will be sent into the DCNN model for training purposes in the second stage. In the third stage, the preprocessed LR spectrum data will be fed into the trained model with the aid of the optimized hyperparameters, and the trained DCNN can generate the HR spectrum data. Additionally, performance results show that the proposed reconstruction scheme can obtain the reconstructed HR spectrum data in terms of the low mean absolute error.

Keywords: Spectrum data reconstruction · Down-sampling · Deep convolutional neural network

1 Introduction

Satellite spectrum sensing has the advantages of wide sensing range, efficient spectrum utilization and low energy consumption, which has received increasing research attention as a benefit of its ability of seamless coverage [1, 2]. With the increasing number of sensing tasks, the amount of data transmitted by the satellite spectrum sensing system become enormous. However, these huge amount of data may not be transmitted, due to the limited transmission capability of wireless links spanning from satellites to earth stations [3].

At present, there are two kind ways to solve the problem of limited data transmission, which are data compression and down-sampling. To be specific, data compression [4] relies on high timely processing ability on-satellite. In contrast, down-sampling is more easier to implement on-satellite, and the unsampled data can be reconstructed at the Earth station. The data reconstruct methods are mainly concentrated in the field of image processing, such as super-resolution (SR) [5], and image interpolation [6], etc.

Q. Wu et al. (Eds.): WiSATS 2020, LNICST 358, pp. 636–644, 2021.
https://doi.org/10.1007/978-3-030-69072-4_52

Image Super-resolution can reconstruct low-resolution (LR) images as high-resolution (HR) images [7, 8]. Especially, with the continuous development of deep learning [9], the super-resolution Convolutional Neural Network (SRCNN) was designed to recover LR images [10]. To be specific, Bicubic interpolation method [11] was used to preprocess the image, and then sent the preprocessed data into the network to learn the end-to-end features between LR images and HR images, which achieved a better recovery performance compared with the traditional image recovery technology. Due to the adaptability and robustness, in [12], deep convolutional network was utilized to reconstruct LR images.

Inspired by this, the spectrum data with time-frequency domain can also be regarded as an image. Through our investigation and research, there is no deep learning-based method introduced into the spectrum reconstruction. In this paper, we propose a super-resolution reconstruction method for LR spectrum relying on DCNN. Moreover, the Adam optimization [13] is adopted, and the zero interpolation method [14] is also used to improve the efficient of data preprocessing. The main contributions of this paper are as follows: we propose a spectrum-data super-resolution reconstruction with the aid of deep convolutional neural network, thus, the spectrum sensing satellites can perform spectrum sensing relying on down-sampling, and the proposed reconstruction method can generate HR spectrum data. Furthermore, performance evaluations show that the proposed spectrum-data reconstruction scheme has a good reconstruction quality.

2 Data Preprocessing

Similar to image recovery, the LR spectrum data can also be reconstructed based on Convolutional Neural Networks. In the image recovery, the Bicubic interpolation method is an effective way that usually used for preprocessing to obtain the LR samples [11]. In contrast, the spectrum data has two-dimensional nature and correlation in time and space, where time-domain and frequency-domain are equivalent to the height and width of the grayscale image, respectively, and the power spectral density (PSD) value is equivalent to the pixel value. Thus, the spectrum data of the time-frequency domain can be considered as the grayscale image for processing. However, differing form the image, the time-frequency domain interval of spectrum data is inconsistent, and the LR spectrum data should also exist in the HR case. Then, the zero interpolation method is adopted for spectrum data construction, without using Bicubic method.

Real-world spectrum data are collected and stored as PSD value. To reduce the impact of random noise and avoid direct reconstruction of the PSD value, the sensed data should be preprocessed, including normalization and zero interpolation. Then, the normalized HR spectrum data is considered as a HR spectrum image, which can be used as the training label. Moreover, for training purposes, the HR spectrum image can be converted into LR spectrum image via down-sampling. Moreover, the LR spectrum image will be inserted with zero by the factor of d in both time domain and frequency domain. Taking the factor $d = 2$ as an example, the specific zero interpolation can be shown in Fig. 1.

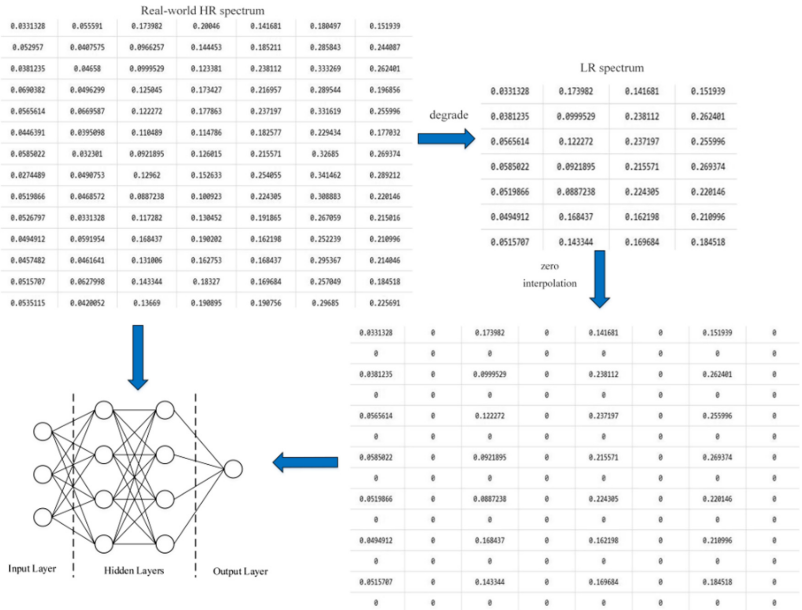

Fig. 1. Data preprocessing in training phase

The data preprocessing in the reconstruction phase only relies on normalizing the acquired LR spectrum data, and carries out zero interpolation operation. Then, the preprocessed can fed into the model to achieve recovery and reconstruction, as shown in Fig. 2.

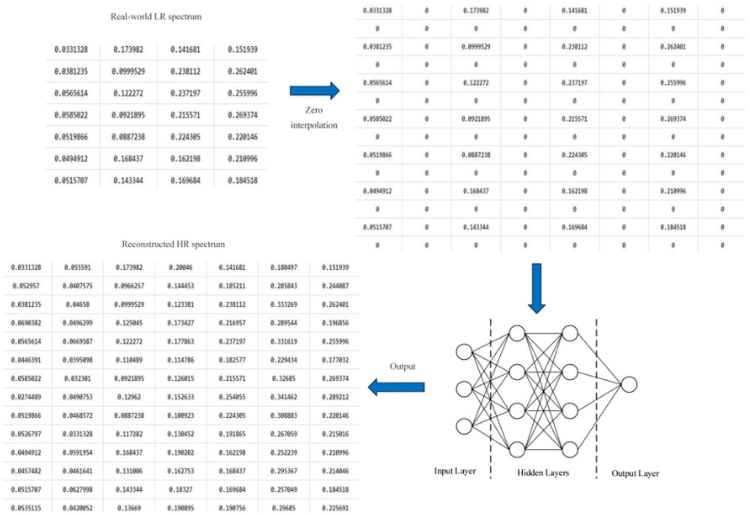

Fig. 2. Data preprocessing in the testing phase

3 Deep Convolutional Neural Networks for Spectrum Data Reconstruction

3.1 Deep Convolutional Neural Networks

Compared with other neural networks, DCNN has the advantages of sparseness, parameter sharing, non-linearity, translation invariance, etc. Moreover, it has advantage of learning the correlation in space [15]. Therefore, it can achieve excellent performance in image processing and classification, leading to extensive research [9]. In order to perform super-resolution reconstruction of the spectrum based on DCNN network, a three-layer neural network is constructed, which can process a large amount of data, and learn an end-to-end feature mapping between LR spectrum and HR spectrum. Moreover, it also can estimate missing HR information in the LR spectrum in order to complete reconstruction of spectrum data, relying on spectrum data preprocessing, spectrum feature extraction, non-linear mapping and final reconstruction. The specific model is shown in Fig. 3:

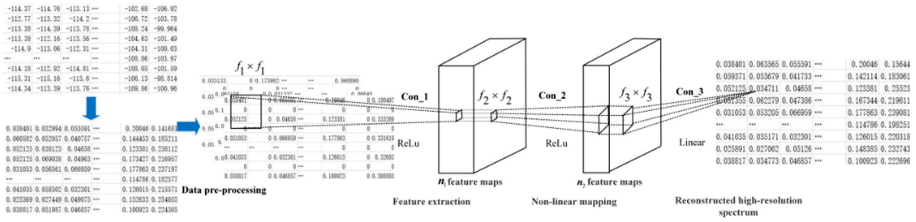

Fig. 3. DCNN-based spectrum super-resolution reconstruction

As shown in Fig. 3, no pooling layer is used to prevent the loss of information. The rectified linear unit (ReLU) has been proven to keep the gradient from decaying and effectively overcome the problem of gradient disappearance and gradient explosion, which is prone to occur during the reverse transfer of the parameters of the model. Moreover, it can also accelerate the model's convergence rate and improve the performance of the convolutional network [16]. Hence, ReLu is used as the activation function of the convolution layer for the hidden layer neuron output, which can be shown as:

$$y = \max(0, x) \tag{1}$$

The preprocessed LR spectrum date, denoted by $\boldsymbol{X}$, will be sent into the constructed DCNN. Each neuron in the first layer consists of both learnable weights and biases, and the features of LR spectrum are extracted and expressed. In particular, the input spectrum $\boldsymbol{X}$ is weighted and summed, and it will be passed to an activation function and be converted into high-dimensional vector. Finally, all these vectors are spatially recombined to output a feature map. Upon denoting this operation as F_1, which can be expressed as:

$$F_1(\boldsymbol{X}) = \max(0, \boldsymbol{W}_1 * \boldsymbol{X} + \boldsymbol{B_1}) \tag{2}$$

where $\boldsymbol{W_1}$ is the set of n_1 convolution kernels, and the size of each convolution kernel in the set is $c \times f_1 \times f_1$. Moreover, $\boldsymbol{B}_1$ represents the bias, and its each element corresponds to a convolution kernel.

The first layer uses n_1 convolution kernels for convolution operations (*), and every convolution kernel outputs a feature map to form a n_1-dimensional feature map. In contrast, the second layer is a non-linear mapping. A convolution kernel of 1×1 size is used. This is due to that it can reduce the number of parameters by cutting back the number of input channels. Moreover, this layer output n_2-dimensional feature map, and the operation process be expressed as:

$$F_2(\boldsymbol{X}) = \max(0, \boldsymbol{W_2} * F_1(\boldsymbol{X}) + \boldsymbol{B_2}) \tag{3}$$

where the weight $\boldsymbol{W_2}$ has n_2 convolution kernels of size $n_1 \times f_2 \times f_2$, and the bias $\boldsymbol{B}_2$ is a n_2-dimensional vector. This convolution layer can be used to perform non-linear mapping, and to obtain the characteristic information of the reconstructed HR spectrum.

The third layer of the convolutional network can be used to reconstruct the high-resolution spectrum, and to perform the average operation to process the data with overlapping areas, resulting in obtaining the final complete spectrum. The average operation convolves the spectrum relying on a mean filter, assigns the average value in the neighborhood to the central element, and removes irrelevant details in the data. The procedure be expressed as:

$$F(\boldsymbol{X}) = \boldsymbol{W}_3 * F_2(\boldsymbol{X}) + \boldsymbol{B_3} \tag{4}$$

where W_3 contains n_3 linear convolution kernels of size $n_2 \times f_3 \times f_3$, and the bias $\boldsymbol{B_3}$ is a n_3-dimensional vector. Finally, a complete reconstructed HR spectrum is obtained. The optimization of related hyper-parameters will be detailed later.

3.2 Loss Function, Optimizer and Evaluation Metric

In order to evaluate the reconstruction performance, the mean-square error (MSE) is used as the loss function of DCNN, which can indicate the distance between the reconstructions and the targets by loss scores, and can be formulated as:

$$L(\theta) = \frac{1}{K} \sum_{k=1}^{K} \| F(X_k; \theta) - Y_k \|^2 \tag{5}$$

where K denotes the total number of spectrum training samples, k is the sample number, $\{\boldsymbol{Y}_k\}$ is a set of real-world HR spectrum, and $\{\boldsymbol{X}_k\}$ is the corresponding set of LR spectrum. The purpose of network training is to update parameters $\boldsymbol{\theta} = \{\boldsymbol{W_1}, \boldsymbol{W_2}, \boldsymbol{W_3}, \boldsymbol{B_1}, \boldsymbol{B_2}, \boldsymbol{B_3}\}$ with input data iteratively, and establish end-to-end feature mapping F between LR spectrum and HR spectrum.

In the SRCNN model [10], a stochastic gradient descent (SGD) optimizer is used to update the network parameters and to minimize the loss function [17]. The SGD algorithm updates the weights with the aid of using the same fixed learning rate set in advance for all parameters. A large number of parameters are often involved in the model,

and the updating frequencies of different parameters are often different. If the learning rate is set too large or too small, it will increase the number of iterations, and slow down the convergence rate. Considering these problems, the Adagrad algorithm determines the updating amplitude by calculating the sum of the squares of all gradients [18], which gradually reduces the learning rate of each parameter. Differing from Adagrad, based on [19], RMSProp algorithm gradually discards past gradients and adds the rest in an exponential moving average manner to achieve a customized learning rate for each parameter. From [13], the adaptive moment estimation (Adam) algorithm uses gradient first moment estimation and second raw moment estimation to dynamically design different adaptive learning rates for each parameter. Thus, Adam algorithm is utilized, which is more suitable for processing spectrum data. Specific experimental results will be evaluated later.

The model reconstruction performance is evaluated with the aid of the mean absolute error (MAE) value [20], which is used to evaluate the normalized PSD error between the real-world HR spectrum and the reconstructed, and can be given by:

$$\mathrm{MAE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| y(i,j) - \hat{y}(i,j) \right| \tag{6}$$

where M is the total number of frequency points, and N is the total time slots. $y(i,j)$ is the normalized PSD that i-th frequency point at the j moment in the real-world HR spectrum, $\hat{y}(i,j)$ is the normalized PSD of the reconstructed spectrum correspondingly. i and j are the frequency point and time slot, respectively.

4 Experiments and Evaluations

4.1 Experimental Precondition

The experiment platform is Inter(R) Core(TM) i7-9700K CPU @ 3.60 GHz, GPU is NVIDIA RTX2080ti, memory is 64.00 GB, and convolutional neural network is built on the Tensorflow framework. The dataset used in the experiment is the spectrum data of the GSM1800 downlink, ranging from 1820 to 1875 MHz, and the time span is two weeks. The minimum sampling interval in the frequency domain of the dataset is 200 kHz, and the minimum interval in the time domain is 1.8 s. To save computing resources and reduce data redundancy, we resample the original dataset at 18 s interval.

4.2 Selection of Network Hyper-Parameters

Hyper-parameters (network depth, convolution kernel size and the number, etc.) are important factors, which can affect the performance of model. The specific selection process can be shown as follows. First, the width of the network is changed by the number of convolution kernels, which means that the width of the network refers to the number of convolution kernels per layer. In order to output the complete reconstructed spectrum, the number of convolution kernel in the last layer is fixed to 1. Four groups of different convolution kernel number combinations are selected. From Table 1, it is

clear that the performance of the network improves as the network width increases. But considering the time overhead, the wider the network width costs more time. Hence, we have $n_1 = 128, n_2 = 64, n_3 = 1$.

Table 1. Evaluation of different convolution kernel combinations

n_1	n_2	Total params	Time ($\mu s/step$)	Test MAE
256	128	57089	93	0.0258
128	**64**	**20353**	**55**	**0.0261**
64	32	8129	38	0.0266
32	16	3353	29	0.0271

Second, different convolution kernel sizes are also evaluated. The convolution kernel size of the second layer is set to 1×1, which adds non-linear features without loss of resolution, improves the network expression ability and reduces the computational complexity [21]. From Table 2, we have $f_1 = 9, f_2 = 1, f_3 = 5$.

Table 2. Evaluation of different convolution kernel sizes

f_1	f_2	f_3	Total params	Time ($\mu s/step$)	Test MAE
11	1	9	29057	80	0.0260
9	**1**	**5**	**20353**	**55**	**0.0261**
7	1	3	15233	52	0.0267

Finally, the effect of the model depth is analyzed. The model parameters are shown in Table 3. The comparison results of the reconstruction performance of the model are shown in Fig. 4. It can be seen that the convergence speed of the three-layers network is slightly faster than that of the four-layers network, and the three-layers network has a better performance than that of the four-layer network. Therefore, the selection of model depth is 9-1-5.

Table 3. Model parameter settings

Network depth	Network composition	Hyper-parameter combination
3	**9-1-5**	$\boldsymbol{n_1 = 128, n_2 = 64, n_3 = 1}$
4	9-1-1-5	$n_1 = 128, n_2 = 64, n_3 = 32, n_4 = 1$
4	9-1-1-5	$n_1 = 128, n_2 = 64, n_3 = 16, n_4 = 1$
4	9-7-1-5	$n_1 = 128, n_2 = 64, n_3 = 32, n_4 = 1$

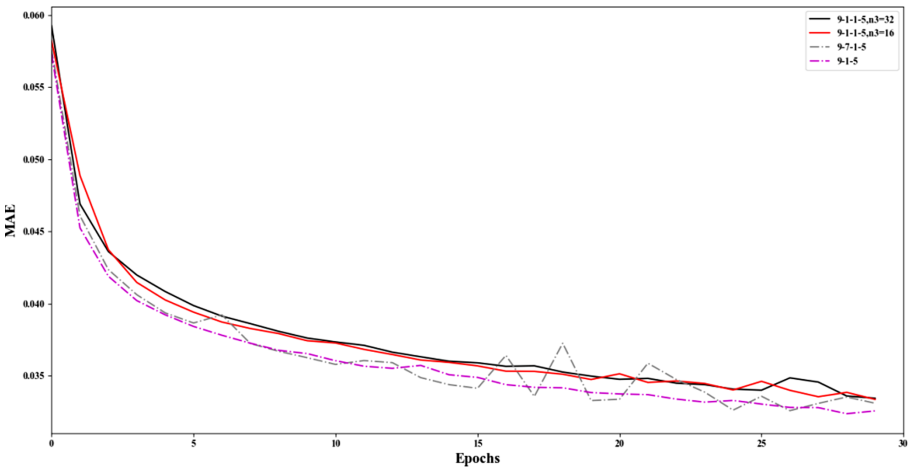

Fig. 4. MAE values for Layer 3 and Layer 4 networks

5 Conclusion

In this paper, we investigated the reconstruction of LR spectrum data. In order to reconstruct LR data, a DCNN-based spectrum super-resolution scheme was designed. Specifically, in the training phase, a small amount of HR spectrum data may be preprocessed, and the preprocessed data will be fed into the designed DCNN model. Then, the model can mine correlation between the LR spectrum and the HR spectrum in both time domain and frequency domain. Moreover, in the testing phase, the received LR data is normalized, and sent into the model, then the model can output the generated HR spectrum data. Experimental results have demonstrated that the proposed reconstruction scheme can achieve good reconstruction performance.

Acknowledgement. This work presented was partially supported by the National Science Foundation of China (No. 91738201), the China Postdoctoral Science Foundation (No. 2018M632347), and the Natural Science Foundation for Jiangsu Higher Education Institutions (No. 18KJB510030).

References

1. Abdelmohsen, A., Hamouda, W.: Advances on spectrum sensing for cognitive radio networks: theory and applications. IEEE Commun. Surv. Tutor. **19**(2), 1277–1304 (2016)
2. Zhang, L., et al.: A survey of advanced techniques for spectrum sharing in 5G networks. IEEE Wirel. Commun. **24**(5), 44–51 (2017)
3. Jia, M., et al.: Broadband hybrid satellite-terrestrial communication systems based on cognitive radio toward 5G. IEEE Wirel. Commun. **23**(6), 96–106 (2016)
4. Ghahremani, M., Ghassemian, H.: A compressed-sensing-based pan-sharpening method for spectral distortion reduction. IEEE Trans. Geosci. Remote Sens. **54**(4), 2194–2206 (2015)
5. Niu, X.: An overview of image super-resolution reconstruction algorithm. In: 2018 11th International Symposium on Computational Intelligence and Design (ISCID), China, pp. 16–18 (2018)
6. Huang, J., Siu, W., Liu, T.: Fast image interpolation via random forests. IEEE Trans. Image Process. **24**(10), 3232–3245 (2015)
7. Farsiu, S., et al.: Advances and challenges in super-resolution. Int. J. Imaging Syst. Technol. **14**(2), 47–57 (2014)
8. Tsai, R.: Multiframe image restoration and registration. Adv. Comput. Visual Image Process. **1**, 317–339 (1984)
9. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
10. Dong, C., et al.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 295–307 (2015)
11. Wang, H., Zhou, L., Zhang, J.: Region-based bicubic image interpolation algorithm. Comput. Eng. **19**, 216–218 (2010)
12. Liu, W., et al.: A survey of deep neural network architectures and their applications. Neurocomputing **234**, 11–26 (2017)
13. Mehta, S., Paunwala, C., Vaidya, B.: CNN based traffic sign classification using Adam optimizer. In: 2019 International Conference on Intelligent Computing and Control Systems (ICCS), India, pp. 1293–1298 (2019)
14. Ionutiu, R., Rommes, J., Antoulas, A.: Passivity-preserving model reduction using dominant spectral-zero interpolation. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **27**(12), 2250–2263 (2008)
15. Hu, W., et al.: Deep convolutional neural networks for hyperspectral image classification. J. Sens. **2015**(258619), 1–12 (2015)
16. Nair, V., Hinton, G.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), Israel, pp. 807–814 (2010)
17. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT 2010 Physica-Verlag HD, Paris, pp. 177–186 (2010)
18. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
19. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA: Neural Netw. Mach. Learn. **4**(2), 26–31 (2012)
20. Willmott, C., Matsuura, K.: Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Res. **30**(1), 79–82 (2005)
21. Pang, Y., et al.: Convolution in convolution for network in network. IEEE Trans. Neural Netw. Learn. Syst. **29**(5), 1587–1597 (2018)