DAE+CNN-Based Radiation Source Signal Recognition Algorithm (Postprint)

Authors: Ye Wenqiang, Yu Zhifu, Zhang Kui

Date: 2018-10-11T00:00:00+00:00

Abstract

This paper investigates the high time complexity problem in emitter signal recognition using convolutional neural networks and proposes an algorithm that combines denoising autoencoders with convolutional neural networks. First, short-time Fourier transform is applied to radar emitter signals to obtain time-frequency images; then, the images undergo grayscale conversion and threshold binarization processing. The processed images are vectorized and input into a denoising autoencoder, where hidden layer feature data are extracted to accomplish dimensionality reduction. These features are subsequently reconstructed into image matrices and fed into a convolutional neural network for classification and recognition using the standard Softmax classifier. Simulation results demonstrate that the model incorporating denoising autoencoder-based dimensionality reduction achieves a substantial reduction in time complexity compared to the original model, attaining recognition performance exceeding 80% at SNR = -6 dB, with significantly improved recognition performance compared to conventional dimensionality reduction methods.

Full Text

Preamble

Recognition Algorithm of Emitter Signal Based on DAE+CNN

Authors: Ye Wenqiang, Yu Zhifu, Zhang Kui (Institute of Electronic Countermeasure, National University of Defense Technology, Hefei 230037, China)

Abstract: This paper addresses the high time complexity problem in emitter signal recognition using convolutional neural networks (CNNs) and proposes a novel algorithm combining denoising auto-encoders (DAE) with CNNs. The approach first performs short-time Fourier transform (STFT) on radar emitter signals to obtain time-frequency images, which are then converted to grayscale and thresholded to binary. The processed images undergo vectorization and are fed into a denoising auto-encoder to extract hidden layer features for dimensionality reduction. The reduced-dimensional data is subsequently reshaped into image matrices and input into a CNN for classification using a standard Softmax classifier. Simulation results demonstrate that the proposed DAE-enhanced model significantly reduces time complexity compared to the original CNN architecture while achieving over 80% recognition accuracy at SNR = -6 dB. The recognition performance substantially outperforms traditional dimensionality reduction methods.

Keywords: radar emitter; short-time Fourier transform (STFT); denoising auto-encoder (DAE); convolutional neural network (CNN); softmax

0 Introduction

Radar emitter signal recognition plays an irreplaceable role in modern warfare, directly affecting the performance of electronic reconnaissance equipment and critically influencing subsequent operational decision-making. It not only enables signal reconnaissance but also determines whether enemy weapons pose a threat, making it strategically significant to battlefield outcomes. As warfare evolves and technology advances, battlefield requirements for radar emitter signal recognition have become increasingly demanding. Traditional manual identification methods can no longer satisfy modern warfare needs, necessitating the introduction of artificial intelligence and intelligent information processing techniques [?].

Machine learning methods can achieve excellent recognition performance for radar emitter signals but require massive training datasets, often leading to the curse of dimensionality and increased model time complexity. Dimensionality reduction is the most common solution, mapping data from high-dimensional to low-dimensional space through methods such as PCA, LDA, and LLE [?]. However, these traditional approaches only preserve single features, and using their reduced-dimensional data as input for deep learning models significantly impacts recognition performance. In 2006, Hinton published in *Science* that multiple hidden layers could be used to achieve high-to-low dimensional encoding, with the learned low-dimensional representation being closer to the essential nature of the original data [?].

Previous research has explored various deep learning approaches. Reference [?] utilized stacked sparse auto-encoders for conventional radar emitter signal recognition, achieving significantly improved recognition compared to traditional methods but suffering from high time complexity. Reference [?] applied joint PCA and random projection dimensionality reduction before using stacked sparse auto-encoders, which reduced time complexity but substantially decreased recognition performance. Reference [?] employed CNNs for recognition by improving activation functions and kernel sizes, achieving better loss function convergence and good recognition results, but without preprocessing bispectrum images, the excessive data dimensionality resulted in overly complex network structures.

Building upon these foundations, this paper proposes integrating a denoising auto-encoder for preprocessing into an established CNN framework. The DAE performs dimensionality reduction through its hidden layers before feeding the data into the CNN for recognition, thereby addressing the time complexity issue while maintaining recognition performance.

1.1 Denoising Auto-Encoder

Denoising auto-encoders (DAE), proposed by Vincent et al. in 2010 [**?**], achieve superior feature learning by corrupting input data with noise. An auto-encoder (AE) is an unsupervised three-layer neural network comprising an input layer, hidden layer, and output layer [**?**], with training involving encoding and decoding processes. The network structure is illustrated in [Figure 1: see original paper].

The encoding process maps the input vector to the hidden layer to obtain a new feature representation:

$$f^{(1)}(z) = \sigma(W^{(1)}x + b^{(1)})$$

where $x \in \mathbb{R}^{n\times 1}$ is the input vector, $W^{(1)} \in \mathbb{R}^{m\times n}$ is the hidden layer weight matrix, $b^{(1)} \in \mathbb{R}^{m\times 1}$ is the hidden layer bias, z is the output vector, and σ is the activation function. Common activation functions include the Sigmoid function:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

and the tanh function:

$$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

The decoding process maps the hidden layer back to the input space:

$$g^{(2)}(s) = \sigma(W^{(2)}z + b^{(2)})$$

where $W^{(2)} \in \mathbb{R}^{n\times m}$ and $b^{(2)} \in \mathbb{R}^{1\times n}$.

Combining equations (1) and (2), the reconstruction error is:

$$L_D = \|g(f(x)) - x\|^2$$

The cost function is defined as:

$$J(W,b) = \frac{1}{2N}\sum_{i=1}^{N}\|g(f(x^{(i)})) - x^{(i)}\|^2 + \frac{\lambda}{2}\sum_{l=1}^{2}\sum_{i=1}^{S_l}\sum_{j=1}^{S_{l+1}}(W_{ji}^{(l)})^2$$

where $x^{(i)}$ represents the i-th sample, $W_{ji}^{(l)}$ denotes the connection weight between unit i in layer l and unit j in layer $l+1$, N is the number of samples, S_l is the number of units in layer l, and λ is the regularization coefficient (set to 1 in this paper).

Optimal solutions for W and b can be obtained using error backpropagation and batch gradient descent algorithms [?]. By adding noise to training data, the DAE is forced to learn noise removal. To recover uncorrupted input data [?], the DAE can find more stable and deeper features under noisy conditions, which constitute a higher-level representation of the input data and significantly improve overall model stability. The denoising training principle is shown in [Figure 2: see original paper], where x represents original input data, $\tilde{x}$ is the corrupted input, z encodes $\tilde{x}$, and $\hat{x}$ decodes z. The reconstruction error is $L_D = \|g(f(\tilde{x})) - x\|^2$.

1.2 Convolutional Neural Network

Convolutional neural networks (CNNs) are specialized for processing data with grid-like topology [?], such as time-series data (one-dimensional grids formed by regular sampling along the time axis) and image data (two-dimensional pixel grids). CNNs have demonstrated exceptional performance across numerous applications. Convolution represents a special linear operation.

Three key concepts improve and optimize CNN learning systems: sparse interaction, parameter sharing, and equivariant representation [?]. Additionally, convolution provides a method for handling variable-sized inputs.

In convolutional layers, the input image or feature map undergoes convolution operations with kernels. The kernel "slides" across the image with a fixed stride, performing convolution at each position. One kernel produces one feature map, while n kernels generate n feature maps. The convolution operation is calculated as:

$$f^{(l)}(x) = \sigma\left(\sum_{j\in M_j} K_{ij}^{(l-1)} * X_j^{(l-1)} + b_i^{(l)}\right)$$

where l represents the layer index in the CNN, K is the convolution kernel, M_j is the receptive field, and b is the bias term.

Pooling/sampling aims to reduce matrix dimensions and computational complexity. Common pooling methods include max pooling, average pooling, L2 norm pooling, and weighted average pooling based on distance from central pixels [?]. The pooling calculation is:

$$x_j^{(l)} = \text{downsample}(X_j^{(l-1)}) + b_j^{(l)}$$

The basic CNN framework is shown in [Figure 3: see original paper]. As a deep feedforward artificial neural network, CNNs have achieved tremendous success in image classification and recognition, comprising convolutional layers, pooling layers, and fully connected layers. Images are transformed through these specialized layers into specific outputs, with convolutional and pooling layers being central to the architecture. Their parameter settings critically impact overall network recognition performance.

1.3 Softmax Classifier

Mainstream classifiers include Softmax, SVM, and Bayesian classifiers [?]. This paper selects the Softmax classifier for multi-class problems where sample class labels y can take k values ($k > 2$). Given a training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ where $y^{(i)} \in \{1, 2, \ldots, k\}$, Softmax is a supervised learning algorithm requiring clearly defined sample categories (one sample cannot belong to multiple categories simultaneously). For input x, the probability of belonging to class j is:

$$p(y = j|x) = \frac{e^{\theta_j^T x}}{\sum_{l=1}^{k} e^{\theta_l^T x}}$$

where $\theta = [\theta_1, \theta_2, \ldots, \theta_k]^T$ and $\theta_i \in \mathbb{R}^{n+1}$. The corresponding neural network hypothesis function is:

$$h_\theta(x) = \begin{bmatrix} p(y = 1|x;\theta) \\ p(y = 2|x;\theta) \\ \vdots \\ p(y = k|x;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x}} \begin{bmatrix} e^{\theta_1^T x} \\ e^{\theta_2^T x} \\ \vdots \\ e^{\theta_k^T x} \end{bmatrix}$$

The normalization term ensures output probabilities sum to 1 [?]. Since radar emitter signal time-frequency features exhibit minimal differences, Softmax can find weight values that amplify subtle inter-sample distinctions, providing finer classification. While SVM classifiers are primarily used for samples with larger differences, Softmax extends logistic regression from binary to k-class problems, calculating output probabilities for each category with simple computation. Therefore, this paper adopts the Softmax classifier.

---

2 Implementation of DAE+CNN-Based Emitter Signal Recognition

2.1 Image Preprocessing

Using the LFM signal $s(t)$ as an example, we apply a Hamming window and perform short-time Fourier transform to obtain $S(t,f)$, as shown in [Figure 4: see original paper]. The time-frequency image cannot be directly input into the DAE network and requires preprocessing.

First, each point in the time-frequency image undergoes normalization:

$$\alpha(t,f) = \frac{S(t,f)}{\max(S(t,f))}$$

The image is then converted to grayscale, as shown in [Figure 5: see original paper]. To remove noise and reduce its impact on subsequent recognition, grayscale images undergo threshold binarization using automatic threshold selection:

$$\alpha(t,f) = \begin{cases} 1, & \alpha(t,f) \geq \text{thred} \\ 0, & \alpha(t,f) < \text{thred} \end{cases}$$

The processed time-frequency image is vectorized into vector $V = [\alpha_0, \alpha_1, \ldots, \alpha_n]^T$, which serves as input to the network model. For n input samples, we represent them as $V = [V_1, V_2, \ldots, V_n]^T$.

2.2 DAE Dimensionality Reduction

DAE dimensionality reduction leverages hidden layer neurons for data compression, preserving original data features as much as possible through compressed representations. The output dimension after the hidden layer equals the number of hidden neurons.

For input samples $V \in \mathbb{R}^{n\times m}$ with individual sample dimension m and n input samples, if the first DAE hidden layer contains h neurons with bias term $b \in \mathbb{R}^{1\times h}$ and weight matrix $W \in \mathbb{R}^{m\times h}$, the feature output is $Z = f(VW + b) \in \mathbb{R}^{n\times h}$. This reduces data from original dimension m to h (the number of hidden neurons).

Equations (13) and (1) differ because sample data rows and columns are transposed. In experiments, row count n represents the number of samples while column count m represents individual sample dimension. Since bias b cannot be directly added due to dimension mismatch, we prepend a column of ones to the input samples, making the input $V \in \mathbb{R}^{n\times(m+1)}$ and weight matrix $W \in \mathbb{R}^{(m+1)\times h}$. The output becomes $Z = f(VW) \in \mathbb{R}^{n\times h}$, enabling dimensionality reduction while avoiding computationally expensive bias replication.

2.3 Algorithm Recognition Flow

The proposed algorithm framework, illustrated in [Figure 6: see original paper], operates as follows:

a) **Data Preprocessing**: Perform STFT on radar emitter signals to obtain time-frequency images. Process these images through normalization, grayscale conversion, and threshold binarization, followed by vectorization to obtain input sample V.

b) **DAE Processing**: Input V into the DAE network, adjust network parameters to determine appropriate architecture, and output feature matrix H.

c) **CNN Processing**: Reshape H into image matrices and input into the CNN, determining suitable CNN parameters.

d) **Classification**: Combine the deep learning model with the recognition task, using the final Softmax classifier for classification and recognition, outputting identification results for each signal type.

---

3 Simulation Experiments

To verify the effectiveness and accuracy of the proposed algorithm, simulations were conducted using common radar emitter signals: continuous wave (CW), binary phase-coded (BPSK), binary frequency-coded (BFSK), quadruple frequency-coded (QFSK), linear frequency-modulated (LFM), and nonlinear frequency-modulated (NLFM). Signal parameters were set as follows: carrier frequency 200 MHz, sampling rate 2 GHz. BPSK and BFSK used 11-bit Barker codes, QFSK used 16-bit Frank code. For each signal type, 200 samples were generated every 2 dB in the SNR range of -6 dB to 4 dB as test sets, and 500 random samples were generated in the range $[0, 10]$ dB as training sets, totaling 3,000 samples across six signal types.

Experiment 1: CNN Baseline

Using CNN as the deep learning structure, network parameters were established: two convolutional layers (first layer: 12 kernels, second layer: 6 kernels, all 5×5), pooling layers of 2×2, learning rate 1.0, and average pooling. Input was a 40×40 matrix. Recognition results across varying SNR are shown in .

Analysis of reveals that standalone CNN recognition achieves excellent results, with recognition rates exceeding 90% at SNR = -6 dB. However, time complexity is high at 21.75 seconds per iteration. Since deep learning requires multiple iterations, this time cost limits practical applicability, making complexity reduction essential.

Experiment 2: Dimensionality Reduction Comparison

Input data was processed through the DAE dimensionality reduction model with a two-hidden-layer architecture (1400 and 1024 nodes). The 3000×1024 output from the second hidden layer was reshaped to $32 \times 32 \times 3000$ and input into the same CNN structure. For comparison, another experiment applied PCA, random projection (RP), and joint PCA+RP [?] for dimensionality reduction to the same $32 \times 32 \times 3000$ format. Results are shown in .

demonstrates that DAE+CNN reduces time by approximately 8 seconds per iteration compared to standalone CNN due to substantially reduced input dimensionality. Since deep learning training requires many iterations, this yields significant time savings. While DAE+CNN recognition performance decreases slightly compared to CNN because dimensionality reduction inevitably causes some feature loss, the performance is essentially equivalent at SNR ≥ -2 dB. Considering both time complexity and recognition performance, DAE+CNN offers greater practical value. Among PCA, RP, and PCA+RP methods, joint PCA+RP performs best by preserving both PCA' s energy features and RP' s subspace distortion characteristics, but still shows large performance gaps compared to DAE, which can maintain multiple features—more features facilitate better CNN training and recognition.

Experiment 3: Dimensionality Impact

Using DAE with two hidden layers (1400 and 625 nodes), input data was reduced to 625 dimensions while keeping other conditions constant to investigate dimensional effects, as shown in [Figure 7: see original paper].

[Figure 7: see original paper] analysis shows that data dimension significantly impacts recognition. With the same reduction method but different target dimensions, higher dimensions yield better recognition performance. At high SNR, reducing input dimension causes feature loss; at low SNR, it causes data distortion. However, 625-dimensional input still achieves 100% recognition at SNR = 2 dB.

Experiment 4: Hidden Layer Depth

Previous experiments used two hidden layers for dimensionality reduction. This experiment compares one-layer and three-layer hidden architectures, all extracting data after the 1024-neuron layer before CNN input, with other parameters unchanged. Results are shown in [Figure 8: see original paper].

[Figure 8: see original paper] analysis reveals that three-layer DAE performs worst, while two-layer performs best. Shallow DAE with one hidden layer suffers severe information loss during encoding, yielding large reconstruction errors. However, excessively deep three-layer structures also reduce recognition performance because multiple encoding operations extract increasingly abstract features, losing detailed information and increasing reconstruction error. Therefore, appropriate DAE depth is crucial for optimal recognition.

---

4 Conclusion

This paper leverages the efficient nonlinear dimensionality reduction capability of denoising auto-encoders to improve an established CNN architecture. The DAE removes redundant information while preserving principal features, effectively mapping high-dimensional input to low-dimensional representations. Simulations demonstrate that DAE+CNN substantially reduces time complexity compared to CNN alone, achieving equivalent recognition performance at SNR = -2 dB and significantly outperforming conventional dimensionality reduction methods. However, DAE+CNN performance is less satisfactory at very low SNR. Future work will focus on optimizing network structures to further improve recognition performance under such conditions.

References

[1] Frater M, Ryan M. Electronic warfare for the digitized battlefield [M]. Artech House, Inc. 2001.

[2] Guo Liming, Kou Yunhan, Chen Tao, et al. Low probability of intercept radar signal recognition based on stacked sparse auto-encoder [J]. Journal of Electronics & Information Technology, 2018, 40 (4): 875-881.

[3] Zhou Zhiwen, Huang Gaoming, Gao Jun, et al. Radar emitter recognition algorithm based on deep learning [J]. Journal of Xidian University, 2017, 44 (3): 77-82.

[4] Fang Chen, Xue Zhi. Signal classification method based on complete bispectrum and convolutional neural network [J]. Application Research of Computers, 2018, 35 (12). http://www.arocmag.com/article/02-2018-12-022.html

[5] Yang Jian, Yang Jingyu. Why can LDA be performed in PCA transformed space [J]. Pattern Recognition, 2003, 36 (2): 563-566.

[6] Hinton G E, Salakhutdinov R R, et al. Supporting online material for "reducing the dimensionality of data with neural networks" [J]. Science, 2006, 504 (5786): 504-507.

[7] Pang Yanwei, Yuan Yuan, Li Xuelong. Gabor-based region covariance matrices for face recognition [J]. IEEE Trans on Circuits & Systems for Video Technology, 2008, 18 (7): 989-993.

[8] Zeng Kun, Yu Jun, Wang Ruxin, et al. Coupled deep autoencoder for single image super-resolution [J]. IEEE Trans on Cybernetics, 2017, 47 (1): 27-37.

[9] Zhang Yifei, Chen Zhong, Zhang Feng, et al. Remote sensing image classification based on stacked denoising autoencoder [J]. Journal of Computer Application, 2016, 36 (2): 171-174.

[10] Shao Ling, Cai Ziyun, Li Liu, et al. Performance evaluation of deep feature learning for RGB-D image/video classification [J]. Information Sciences, 2017, 385 (C): 266-283.

[11] Yang Yuhao, Sun Jingming, Yu Shengkang, et al. High Resolution Range Profile Target Recognition Based on Convolution Neural Network [J]. Modern Radar, 2017 (12): 24-28.

[12] Zhang Kai, Zuo Wangmeng, Chen Yunjin, et al. Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising [J]. IEEE Trans on Image Processing, 2017, 26 (7): 3142-3155.

[13] Niu Xiaoxiao, Suen Ching. A novel hybrid CNN-SVM classifier for recognizing handwritten digits [J]. Pattern Recognition, 2012, 45 (4): 1318-1325.

[14] Chen Jinnian, Huang Houkuan, Tian Fengzhan, et al. Effective Classifier for Incomplete Data [J]. Computer Science, 2008, 35 (9): 162-164.

[15] Liu Yachong, Tang Zhiling. Classification and identification method of communication radiation source feature based on softmax regression [J]. Computer Engineering, 2018, 44 (2): 98-102.

Note: Figure translations are in progress. See original paper for figures.