A Novel Lidar Signal Denoising Method Based on Convolutional Autoencoding Deep Learning Neural Network

Minghuan Hu, Jiandong Mao, Juan Li, Qiang Wang and Yi Zhang

Special Issue
Atmospheric Aerosol Optical Properties
Edited by
Dr. Sandra Mogo, Dr. Edith Rodríguez, Dr. Natalia Prats and Dr. Boris Barja
A Novel Lidar Signal Denoising Method Based on Convolutional Autoencoding Deep Learning Neural Network

Minghuan Hu 1,2, Jiandong Mao 1,2,* , Juan Li 1,2, Qiang Wang 1,2 and Yi Zhang 1,2

1 School of Electrical and Information Engineering, North Minzu University, North Wenchang Road, Yinchuan 750021, China; 20197220@stu.nun.edu.cn (M.H.); 2004046@nmu.edu.cn (J.L.); 2013057@nmu.edu.cn (Q.W.); 2013091@nmu.edu.cn (Y.Z.)
2 Key Laboratory of Atmospheric Environment Remote Sensing of Ningxia Province, North Wenchang Road, Yinchuan 750021, China
* Correspondence: maojiandong@nmu.edu.cn

Abstract: The lidar is susceptible to the dark current of the detector and the background light during the measuring process, which results in a significant amount of noise in the lidar return signal. To reduce noise, a novel denoising method based on the convolutional autoencoding deep-learning neural network is proposed. After the convolutional neural network was constructed to learn the deep features of lidar signal, the signal details were reconstructed by decoding part to obtain the denoised signal. To verify the feasibility of the proposed method, both the simulated signals and the actually measured signals by Mie-scattering lidar were denoised. Some comparisons with the wavelet threshold denoising method and the variational modal decomposition denoising method were performed. The results show the denoising effect of the proposed method was significantly better than the other two methods. The proposed method can eliminate complex noise in the lidar signal while retaining the complete details of the signal.

Keywords: lidar; deep learning; autoencoder; convolutional neural network; denoising

1. Introduction

Because the laser as a light source has the characteristics of good monochromaticity, strong coherence, and high collimation, lidar technology has developed rapidly. As an active measurement method, lidar has been widely used in atmospheric remote sensing and environmental monitoring due to its high temporal and spatial resolution. In particular, it has made important progress in the fine detection of atmospheric aerosol optical properties, microphysical properties, atmospheric temperature, relative humidity and other parameters, and has become an important tool for the study of atmospheric environmental parameters and their spatial-temporal evolution. However, in the actual detection process, the lidar return signal is greatly affected by noise. As the detection range increases, the return signal strength becomes weaker and weaker, and the far-field signal is easily submerged in noise [1]. Therefore, it is of great importance to reduce the noise of the return signal.

For denoising lidar return signals, many methods have been performed. In traditional signal processing, the Fourier transform has been widely used in signal denoising. This method separates out the useful signal according to the principle that the frequency of the useful signal is lower than that of the noise, and it can be effective for the processing of linear stationary signals. However, the lidar signal is a nonlinear, nonstationary signal. If it is processed by traditional Fourier transform, it will cause significant distortion [2], thus making the noise-reduction effect unsatisfactory.

The wavelet transform can overcome the shortcomings of the traditional Fourier transform, and has been widely used in signal noise reduction. When using wavelet transform to denoise data, the signal will be divided into a low frequency part and high frequency part.
Useful signals are mostly concentrated in the low frequency part, and the high frequency part is considered to be noise, so only the low frequency part is restructured as useful signals. In this way, the denoising effect is better, but some useful signal components of the high frequency part are ignored, which can also easily cause significant distortion. In 2011, on the basis of wavelet denoising algorithm, J. Mao et al. used wavelet packet analysis method to denoise the lidar signal by restructuring the high and low frequency component, thereby inverting a more accurate extinction coefficient profile [3]. In 2016, X. Qin et al. employed an adaptive method combining wavelet analysis and neural networks to denoise lidar return signals [4], which combined the advantages of wavelet analysis and neural networks to obtain better denoising effects.

In recent years, empirical mode decomposition (EMD) algorithms have also been applied in lidar signal denoising. In 1998, Huang et al. proposed the Hilbert–Huang transform method, including EMD and Hilbert analysis, and decomposed the signal characteristics into different eigenmode functions layer by layer [5]. In 2009, F. Zheng et al. tried to apply the EMD algorithm to lidar signal filtering, achieving remarkable results [6]. In 2020, X. Cheng et al. proposed a denoising method based on ensemble empirical mode decomposition (EEMD), combining segmented singular value decomposition and lifting wavelet transform, which is more suitable for the denoising of lidar return signals [7]. Moreover, in 2014, Konstantin Dragomire et al. proposed the variational modal decomposition (VMD) method, which has obvious advantages in processing nonlinear and nonstationary signals [8]. In 2018, F. Xu et al. successfully applied VMD to denoise lidar return signals, and the effect was significantly better than that of wavelet analysis and other methods [9].

In addition to these studies, there have been studies on noise reduction of photon counting lidar signals, such as the use of Poisson distribution. In 2016, a new method was proposed by W. Marais et al. to solve the problem of effective inversion of high-resolution and lower signal-to-noise ratio observations in nonuniform scenes. By using spatial and temporal correlation in the image and the Poisson distributed noise model, the inversion results were better and precisely maintained the spatial and temporal resolution, while significantly reducing the noise [10]. In 2017, they considered the denoising and reconstruction of images corrupted by Poisson noise, proposed a regularized maximum likelihood formula for reconstruction of Poisson images, and proved that this formula can be solved by a coarse-to-fine proximal gradient optimization algorithm and is easier to generalize to inverse problem settings compared to the BM3D denoising method [11]. In 2020, targeting the problem of suppressing random noise of photon-counting lidar signals, M. Hayman et al. introduced the Poisson refinement to generate statistically independent profiles from photon counting data, which can make the best adjustment to signal processing and optimize the smooth core of a specific photon-counting scene to achieve optimal filtering effects [12].

In recent years, deep learning has been widely used in speech recognition, computer vision, natural language processing and other fields as a popular technology [13]. The deep learning methods are also used in the lidar fields. In 2016, G. Jorge et al. carried out a preliminary study on the applicability of deep learning to improve biomass estimation based on lidar, and the results showed that the autoencoder statistically improved the quality of multi linear regression estimation [14]. In 2020, S. Jennifer et al. used machine learning methods to detect boundary layer heights from the backscattered signal of lidar [15]. In 2021, A. Andreas et al. proposed a new data-driven lidar waveform processing method, which extracted depth information using convolutional neural networks to generate realistic waveform data sets based on specific experimental parameters or large-scale synthesis scenes [16].

Although researchers have developed a large number of denoising methods, few of them applied the deep learning methods for denoising lidar signals. At present, the wavelet neural network algorithm is used for signal denoising, but such an adaptive denoising algorithm often needs a given tutor signal. But in actual lidar measurement, it is difficult for signal denoising to give a tutor signal. Therefore, an adaptive deep learning
denoising method that does not require a given tutor signal has good potential. As an unsupervised learning algorithm, the autoencoder meets the requirements very well. As a learning method for feature extraction and data dimensionality reduction, the autoencoder is a branch of neural network consisting of two parts: encoding and decoding [17]. The encoding part sparsely expresses the input data, and the decoding part completes the reconstruction of the data. The autoencoder developed from the initial data dimensionality reduction method into a data generation model, and ultimately evolved into several models, such as the denoising autoencoder, the sparse autoencoder, the convolutional autoencoder, the contraction autoencoder, the variational autoencoder and so on [18]. Among them, the convolutional autoencoder was proposed by Masci et al. in 2011 to build convolutional neural networks [19]. On the basis of retaining the advantages of traditional autoencoders, convolutional autoencoders combine the advantages of strong feature-extraction capabilities of convolutional neural networks, which are more suitable for feature extraction with lidar return signals.

In this paper, a method based on convolutional autoencoding neural networks (CAENN) was proposed for denoising the lidar return signal. The method uses the encoding and decoding characteristics of the autoencoder to construct deep learning networks for learning the mapping from noised return signals to clean return signals. A large number of return signals measured by Mie-scattering lidar developed by North Minzu University were used to train the network to realize the automatic features of extraction and denoising of return signals. Several simulations and actual experiments were performed and the feasibility and practicability of the proposed CAENN method were proven by comparison with other methods, including wavelet threshold and the VMD method.

2. Materials and Methods

2.1. Lidar System Equation

In lidar systems, the return signal can be described with the following equation:

\[ P(r) = P_0 C r^{-2} \beta(r) \times e^{-2 \int_0^r \alpha(r)dr} \]  (1)

where \( P(r) \) is the received lidar return signal power at distance \( r \), \( P_0 \) is the laser emission power, and \( C \) is the system calibration constant, which includes the optical loss of the transmitting and receiving system, the effective receiving area of the receiving system and other system constants. \( \beta(r) \) denotes the backscattering coefficient, \( \alpha(r) \) is the extinction coefficient, and both can be divided into an atmospheric molecular part and an aerosol part.

\( \beta(r) \) and \( \alpha(r) \) are two unknown quantities. In general, it is impossible to solve an equation for two unknowns. Therefore, it is necessary to assume a ratio of aerosol extinction coefficient to backscattering coefficient (namely, lidar ratio) for the inversion of aerosol extinction coefficient when using common algorithms, such as the Collis slope method [20], the Klett method [21] and the Fernald method [22].

2.2. Autoencoder

The autoencoder is an unsupervised learning algorithm whose output enables the reproduction of input data [23]. It is composed of two parts of neural network, namely the encoding part and the decoding part. The basic autoencoder can be thought of as a three-layer neural network structure: the input layer, the hidden layer, and the output layer. Figure 1 shows the structure of a basic autoencoder.

In previous neural networks, the input sample was usually labeled, so the parameters of the previous layers could be changed, according to the difference between the current output and the label, until convergence. Figure 2 shows the diagram of the training process with labeled samples. However, if the existing data is unlabeled, the previous method is not applicable. Figure 3 shows a diagram of an unlabeled sample training process. When the input \( x \) is sent to an encoder, an encoded output \( y \) is obtained. Here \( y \) is a representation
of the input, but whether $y$ is the input $x$ is unknown, so a decoder is added, and then a decoded output $z$ is obtained. The output $z$ is compared with the input $x$. If $z$ is very similar to $x$, there is reason to believe that $y$ is reliable. Therefore, by adjusting the parameters of the encoder and decoder, the reconstruction error is minimized. At this time, the first representation of the input $x$ is obtained, that is, $y$. Because the lidar data is unlabeled data, the source of the error is obtained directly by comparing the reconstructed data with the original input.

![Diagram](image1)

**Figure 1.** The structure of a basic autoencoder.

![Diagram](image2)

**Figure 2.** The diagram of the training process with labeled samples.

![Diagram](image3)

**Figure 3.** The diagram of the training process with unlabeled samples.

If the output $y$ of the first layer is obtained, the minimum reconstructed error makes one believe that the output $y$ is the approximation of original input signal $x$. Then there is no difference between the second and first layers of training. The output $y$ of the first layer is regarded as the input signal of the second layer; minimizing the reconstruction error again can obtain both the parameters of the second layer and the encoded output.
of the second layer input, that is, the second expression of the original input information. Other layers can be obtained in the same way. Figure 4 shows the diagram of the stacking process.

\[
h = f_\theta(x) = \delta_1(W^{(1)}x + b^{(1)})
\]

(2)

\[
g = f_\theta(h) = \delta_2(W^{(2)}h + b^{(2)})
\]

(3)

where \(x\) is the input lidar return signal, \(h\) is the main feature of the input data, \(g\) is the reconstructed lidar return signal, \(W^{(1)}\) and \(W^{(2)}\) are the weights of the encoder and decoder, \(b^{(1)}\) and \(b^{(2)}\) are the bias of the encoder and decoder, and \(\delta_1\) and \(\delta_2\) are the activation function of the encoder and decoder. In this paper, both encoder and decoder use the ReLU activation function [24].

2.3. One-Dimensional Convolution

For one-dimensional convolution, assuming that the input is a tensor \(x^l \in \mathbb{R}^{L_l \times D_l}\) and the convolution kernel of the current layer is a tensor \(f^l \in \mathbb{R}^{L_l \times D_l \times N}\), the one-dimensional convolution can be written by

\[
y_{l+1,d} = b_d + \sum_{i=0}^{L_l} \sum_{d'=0}^{D_l} f^l_{i,d,n} \times x^l_{i+1+i,d}
\]

(4)

where \(L\) is the size of the input data, \(D\) is the number of channels, \(N\) is the number of convolutions, \(n\) is the \(n\)th convolution, \(i\) is the coordinate of the value, \(d\) is the \(d\)th channel, and \(b\) is the bias. Convolution results can be represented as the sum of convolution at the appropriate location for all channels. Figure 5 shows the one-dimensional convolution calculation process. In this example the convolution kernel is \(f \in \mathbb{R}^{4 \times 1 \times 1}\) and input is \(x \in \mathbb{R}^{8 \times 1}\). By zero filling on both sides of \(x\) (blue area) a new input \(x' \in \mathbb{R}^{12 \times 1}\) can be obtained; when convolution sliding stride 2, by performing a convolution operation of \(f\) on \(x\), the output \(y \in \mathbb{R}^{5 \times 1}\) will be obtained.
3. Principle of CAENN Algorithm

In this paper, the CAENN algorithm was constructed by combining the autoencoder and the one-dimensional convolutional neural network. Compared with the traditional autoencoder, the use of convolution processing instead of full connection is not only conducive to the extraction of data features, but also makes more effective use of the advantages of feature extraction of the autoencoder. The CAENN algorithm designed in this paper was able to extract the main lidar return signals from a relatively rich lidar data training set through the feature extraction process, layer by layer. Through the encoding-decoding process, the CAENN network was able to obtain the sparse expression of lidar return signals, and the original lidar return signal with a lot of noise was converted to lidar return signals with only effective signals, so as to filter out the noise, and then the clean lidar return signal was obtained by decoding part of reconstruction. Figure 6 shows the block diagram of the CAENN algorithm. Firstly, the lidar return signal is preprocessed by the normalization, and the preprocessed sample data is placed in the autoencoding layer of the CAENN algorithm for convolution and pooling encoding processing. Secondly, the detailed features of the sample data are encoded, and the noise is eliminated at the same time. Finally, using the decoding layer of the convolutional autoencoding network, the up-sampling and deconvolution are carried out. Deconvolution decodes the sampling features of the sample, which can recover the sample details and obtain a clean return signal.

![Figure 6. The block diagram of the CAENN algorithm.](image)

3.1. Data Preprocessing

Since the measured lidar return signal values differ greatly, if they are directly sent to the network for training, it may cause numerical problems that will make the training model converge slowly or even not at all. Moreover, the convolutional neural network is relatively sensitive to the data distribution; if the distribution of training data and test data are different, the network training efficiency will decrease, the convergence will slow down, and the prediction effect will eventually be affected. Therefore, it is necessary to normalize the lidar return signal data before network training. According to the minimum value, after centering, the data \( a \) is scaled by the difference between the maximum value and the minimum value, then the data moves with the minimum unit and will be converged
to [0, 1]. This process is called data normalization, which can be expressed with follow equation:

$$b = \frac{a - \min(a)}{\max(a) - \min(a)}$$  \hspace{1cm} (5)

The data normalization process can ensure that the data has a similar scale, which makes the model converge faster and improves the efficiency of network training.

### 3.2. Encoding Network

The encoding part of the CAENN algorithm is alternately composed of a convolutional layer, an activation function, and a down-sampling layer. The convolutional layer acts as a feature extractor to encode the lidar return signal and eliminate noise at the same time. In the convolutional layer, the convolution kernel of the current layer is convolved with the feature vector of the previous layer, and then the feature map of this layer is formed through the activation function. The output of the convolutional layer can be expressed as

$$H_1 = x_n^l = f \left( \sum_{m \in M_n} x_{n-1}^l \times w_n^{mn} + b_n^l \right)$$  \hspace{1cm} (6)

where $x_n^l$ is the feature vector corresponding to the $n$th convolution kernel of the $l$th convolutional layer, $M_n$ represents the receptive domain of the current neuron, $w_n^{mn}$ represents the $n$th weighting coefficient of the $m$th convolution kernel of the $l$th layer, $b_n^l$ denotes the offset coefficient corresponding to the $n$th convolution kernel of the $l$th layer, and $f$ is a nonlinear function. The encoding layer uses the ReLU activation function, which is defined as

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

The down-sampling layer uses pooling technology to maintain features. Here, the maximum pooling operation is used to make features have scaling, displacement, and invariance. At the same time, the down-sampling layer has the function of secondary feature extraction and is expressed as

$$H_2 = x_n^l = f \left( \beta_n^l \times \text{down}(x_{n-1}^l) + b_n^l \right)$$  \hspace{1cm} (8)

where down (·) means down-sampling, $\beta_n^l$ represents the weighting factor, and $b_n^l$ is the bias factor.

### 3.3. Decoding Network

The decoding part of the CAENN algorithm is alternately composed of up-sampling, the deconvolutional layer and the activation functions. The up-sampling is the reverse operation of down-sampling, that is, the up-pooling operation, which can solve the filter overlap problem in the deconvolutional process. The deconvolution is also called transposed convolution. This is because the forward propagation process of the convolutional layer is the back-propagation process of the deconvolutional layer, and the back-propagation process of the convolutional layer is the forward-propagation process of the deconvolutional layer. If the output features of the convolutional layer are up-sampling, the input features of the deconvolutional layer will be obtained. The up-sampling layer can be expressed as

$$H_3 = x_n^l = f \left( \beta_n^l \otimes \text{up}(x_{n-1}^{l-1}) + b_n^{l-1} \right)$$  \hspace{1cm} (9)

where up (·) denotes up-sampling, $\otimes$ represents the up-sampling operation, $\beta_n^l$ is the weighting factor, and $b_n^{l-1}$ is the bias factor. The output of the $l$th deconvolutional layer can be written as
\[ H_4 = x_{\cdot n}^l = f \left( \sum_{m \in M_n} x_{\cdot n}^{l-1} \otimes w_{mn}^l + b_{\cdot n}^l \right) \]  

(10)

where \( x_{\cdot n}^l \) is the feature vector corresponding to the \( n \)th deconvolution kernel of the \( l \)th convolutional layer, \( M_n \) is the receptive domain of the current neuron, \( \otimes \) is the deconvolution operation, \( w_{mn}^l \) is the \( m \)th weighting coefficient of the \( n \)th deconvolution kernel of the \( l \)th layer, \( b_{\cdot n}^l \) is the offset coefficient corresponding to the \( n \)th deconvolution kernel of the \( l \)th layer, and \( f(\cdot) \) is a nonlinear function. The decoding layer also uses the \( \text{ReLU} \) activation function.

3.4. Model Parameters

In this paper, the CAENN model parameters were set by several comparison experiments, for example, the number of convolutional layers, pooling layers and up-sampling layers, the convolution kernel size, and the step size, and other parameters of the one-dimensional convolutional layer were adjusted. Tables 1–3 shows the SNR and MSE of the comparison experiments including the activation function, the number of convolutional layers and the learning rate. From these tables we can conclude the optimal structure and parameters: it used the \( \text{ReLU} \) activation function, the network structure was composed of three convolutional layers with one down-sampling layer or one up-sampling layer alternately, batch size was set to 128, learning rate was set to 0.001, epochs were set to 500, and early stopping was used, which involved stopping training when the model’s performance on the validation set started to decline, in order to avoid the problem of over-fitting caused by continuing training.

Table 1. The comparison experiment of activation function.

<table>
<thead>
<tr>
<th>Activation Function</th>
<th>Training Time</th>
<th>SNR</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReLU</td>
<td>3 h 8 min</td>
<td>12.4673</td>
<td>4.0178 \times 10^{-7}</td>
</tr>
<tr>
<td>Tanh</td>
<td>3 h 27 min</td>
<td>11.9523</td>
<td>4.5238 \times 10^{-7}</td>
</tr>
</tbody>
</table>

Table 2. The comparison experiment of the number of convolutional layers.

<table>
<thead>
<tr>
<th>Number of Convolutional Layers</th>
<th>SNR</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two layers</td>
<td>11.28405</td>
<td>5.2760 \times 10^{-7}</td>
</tr>
<tr>
<td>Three layers</td>
<td>19.2726</td>
<td>8.3842 \times 10^{-8}</td>
</tr>
<tr>
<td>Four layers</td>
<td>14.6482</td>
<td>2.4316 \times 10^{-7}</td>
</tr>
<tr>
<td>Five layers</td>
<td>13.9446</td>
<td>2.8593 \times 10^{-7}</td>
</tr>
</tbody>
</table>

Table 3. The comparison experiment of the learning rate.

<table>
<thead>
<tr>
<th>Learning Rate</th>
<th>SNR</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>15.7745</td>
<td>1.8753 \times 10^{-7}</td>
</tr>
<tr>
<td>0.01</td>
<td>16.6819</td>
<td>1.5224 \times 10^{-7}</td>
</tr>
<tr>
<td>0.005</td>
<td>17.551</td>
<td>1.2451 \times 10^{-7}</td>
</tr>
<tr>
<td>0.001</td>
<td>17.7745</td>
<td>1.1837 \times 10^{-7}</td>
</tr>
</tbody>
</table>

Table 4 lists the optimal CAENN parameters by comparison experiments. The run time in this experiment was only obtained under the follow computer configuration: Intel(R) Core(TM) i5-10210U CPU, Windows10 operating system, RAM 8 GB, and GeForce MX250. Due to the low computer configuration, it took 5 h 38 m 40.508626 s to train the model, but it only took 2.174187 s to use the trained model to test a set of data.
Table 4. The parameters of CAENN method.

<table>
<thead>
<tr>
<th>Layer (Type)</th>
<th>Output Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input_1(InputLayer)</td>
<td>(None, 8800, 1)</td>
</tr>
<tr>
<td>conv1d(Conv1D)</td>
<td>(None, 8800, 128)</td>
</tr>
<tr>
<td>conv1d_1(Conv1D)</td>
<td>(None, 8800, 64)</td>
</tr>
<tr>
<td>conv1d_2(Conv1D)</td>
<td>(None, 8800, 64)</td>
</tr>
<tr>
<td>max_pooling1d(MaxPooling1D)</td>
<td>(None, 4400, 64)</td>
</tr>
<tr>
<td>conv1d_3(Conv1D)</td>
<td>(None, 4400, 32)</td>
</tr>
<tr>
<td>conv1d_4(Conv1D)</td>
<td>(None, 4400, 32)</td>
</tr>
<tr>
<td>max_pooling1d_1(MaxPooling1D)</td>
<td>(None, 2200, 16)</td>
</tr>
<tr>
<td>conv1d_6(Conv1D)</td>
<td>(None, 2200, 16)</td>
</tr>
<tr>
<td>conv1d_7(Conv1D)</td>
<td>(None, 2200, 32)</td>
</tr>
<tr>
<td>conv1d_8(Conv1D)</td>
<td>(None, 2200, 32)</td>
</tr>
<tr>
<td>up_sampling1d(UpSampling1D)</td>
<td>(None, 4400, 32)</td>
</tr>
<tr>
<td>conv1d_8(Conv1D)</td>
<td>(None, 4400, 64)</td>
</tr>
<tr>
<td>conv1d_10(Conv1D)</td>
<td>(None, 4400, 64)</td>
</tr>
<tr>
<td>conv1d_11(Conv1D)</td>
<td>(None, 4400, 128)</td>
</tr>
<tr>
<td>up_sampling1d_1(UpSampling1D)</td>
<td>(None, 8800, 128)</td>
</tr>
<tr>
<td>conv1d_12(Conv1D)</td>
<td>(None, 8800, 1)</td>
</tr>
</tbody>
</table>

3.5. Adam Optimization Algorithm

In view of the problem we want to address, the mean-square error (MSE) is selected as the loss function by comparison, and the network parameters are estimated by minimizing the MSE loss function. Assuming $x_n$ is the original data, the objective function is designed to optimize the objective function through the original data and the reconstructed data $\hat{x}_n$, where $n$ is the number of training samples, and the loss function is expressed as

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \| x_n - \hat{x}_n \|^2 \quad (11)$$

Stochastic gradient descent (SGD) is one of the most commonly used gradient descent methods for optimizing neural networks. However, there are some problems in this training method, such as the selection of learning rate and the oscillation problem during updating. It is often difficult to select an appropriate learning rate, which intensifies the difficulty of training to minimize the objective function. In this paper, the Adam algorithm is used instead of the stochastic gradient descent method for updating the network parameters. The essence of the Adam optimization algorithm is to combine the first-order momentum with the second-order momentum and then correct the deviation. The Adam operator automatically calculates the appropriate learning rate for each parameter, which solves the problem of the stochastic gradient descent method that makes it difficult to choose the learning rate. When the Adam operator is used to solve the weight of a certain neuron, the update process of the $\theta$ value after the $t$th iteration is written as

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (12)$$
$$n_t = \beta_2 n_{t-1} + (1 - \beta_2) g_t^2 \quad (13)$$

where $g_t$ is the first order derivative, $\beta_1$ and $\beta_2$ are the attenuation factor which controls the exponential attenuation, $m_t$ is the exponential moving average value of the gradient obtained from the first moment of the gradient, and $n_t$ is the square gradient obtained from the second moment of the gradient. At this point, both $m_t$ and $n_t$ are biased estimates, each of which must be corrected to become an unbiased estimate of expectations.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1} \quad (14)$$
The corrected $m_t$ and $n_t$ can dynamically update the learning rate, and the last formula for parameter updates is written as

$$\theta_{t+1} = \theta_t - \frac{\hat{m}_t}{\sqrt{\hat{n}_t} + \epsilon} \eta$$

where, $\eta$ is the learning rate. The Adam optimization algorithm can adaptively adjust the updated step size from the two aspects of the gradient mean and the gradient square, instead of being directly determined by the current gradient. The Adam optimization algorithm initializes the parameter vector, the first-order moment vector and the second-order moment vector at first, and then iteratively updates each part to make the parameter $\theta$ converge.

4. Results and Analysis
4.1. Noise Reduction Effect Evaluation

In order to evaluate the effect of denoising algorithm, both the signal-to-noise ratio (SNR) and $MSE$ are generally adopted for evaluation [25]. The SNR is the ratio relationship between signal and noise, and reflects the noise reduction effect. The higher the value, the better the effect of noise elimination. The $MSE$ represents the relationship between the original signal and the denoised signal, and the smaller the value is, the better the effect is. The SNR and $MSE$ are expressed with follow equations:

$$\text{SNR} = 10 \log \left( \frac{\sum_{i=1}^{N} x_i^2}{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2} \right)$$

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{x}_i)^2$$

where $x_i$ is the original signal, $\hat{x}_i$ is the signal after denoising, and $N$ is the signal length.

4.2. Simulation Verification

The interference noise of the lidar return signal mainly includes background radiation noise, shot noise, dark current noise of the detector, and thermal radiation noise. In order to verify the effectiveness and superiority of the CAENN method, the signal containing gaussian white noise is simulated, also a result comparison of CAENN method with wavelet threshold method and VMD method is performed [26]. In this paper, the Block test signal and Bump test signal proposed by Donoho and Johnstone were superimposed with Gaussian white noise, and the two test signals were denoised by the CAENN, wavelet threshold and VMD methods, respectively [27].

In a real environment, noise is often a complex phenomenon from many different sources. Assuming that real noise is regarded as the sum of many random variables with different probability distributions, and that each random variable is independent, their normalized sum tends to Gaussian distribution with the increase in the number of noise sources according to the Central Limit Theorem. In this paper, the Gaussian white noise with $\text{SNR} = 15$ was added to both the Block test signal and the Bump test signal. The purpose of adding noise with a specified $\text{SNR}$ in the simulation experiment was to quantify and evaluate the effect of the algorithm. The wavelet threshold denoising method used db8 wavelet basis function for nine-layer decomposition; the VMD used three-layer decomposition; the CAENN used a self-made data set for training, batch size was 128, and epoch was set to 500. When the model is not optimized 10 consecutive times, it will end early. Figures 7 and 8 show the simulated noise reduction effects for Block and Bump signals, respectively. It can be seen that the three noise reduction methods all had a certain noise reduction effect. The wavelet threshold method filtered out almost all noise, but the
processed Block signal and the Bump signal produced significant distortions compared
to the unprocessed signal, and this effect was the worst. The VMD method had a better
denoising effect and better signal detail retention, but distortion also appeared to some
extent. The CAENN method had the best denoising effect and retained the signal details
very well.

Figure 7. The simulated noise reduction effects of Block signals. (a) the original block signal, (b) the block signal with
\( SNR = 15 \) noise, (c) the signal processed by the wavelet threshold method, (d) the signal processed by the VMD method,
(e) the signal processed by CAENN method.
Figure 8. The simulated noise reduction effects of Bump signals. (a) the original bump signal, (b) the bump signal with SNR = 15 noise, (c) the signal processed by the wavelet threshold method, (d) the signal processed by the VMD method, (e) the signal processed by CAENN method.

In order to further evaluate the effects of three denoising methods mentioned above, the SNR and MSE of the denoised Block signal and Bump signal were calculated, respectively, when superimposed with Gaussian white noise with different SNRs of 5 dB, 10 dB and 15 dB. Table 5 lists the SNR and MSE after noise reduction by three methods when superimposed with noise with different SNRs.
Table 5. The SNR and MSE after noise reduction by three methods when superimposed with noise with different SNRs.

<table>
<thead>
<tr>
<th>Signal</th>
<th>SNRnoise</th>
<th>Wavelet Threshold</th>
<th>VMD</th>
<th>CAENN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SNR</td>
<td>MSE</td>
<td>SNR</td>
<td>MSE</td>
</tr>
<tr>
<td>Block</td>
<td>5 dB</td>
<td>5.1068</td>
<td>1.6028</td>
<td>8.2578</td>
</tr>
<tr>
<td></td>
<td>10 dB</td>
<td>6.6237</td>
<td>1.0074</td>
<td>12.4210</td>
</tr>
<tr>
<td></td>
<td>15 dB</td>
<td>7.2619</td>
<td>0.8391</td>
<td>15.2248</td>
</tr>
<tr>
<td>Bump</td>
<td>5 dB</td>
<td>6.3831</td>
<td>1.1887</td>
<td>8.1213</td>
</tr>
<tr>
<td></td>
<td>10 dB</td>
<td>8.7042</td>
<td>0.6239</td>
<td>12.5492</td>
</tr>
<tr>
<td></td>
<td>15 dB</td>
<td>10.1135</td>
<td>0.4266</td>
<td>17.3844</td>
</tr>
</tbody>
</table>

It is clear that although the weak noise remains, the SNR of the CAENN noise reduction method is nearly twice that of the other two methods, and the MSE of the CAENN method is less than one-tenth of that of the other two methods.

4.3. Denoising Effect of Actually Measured Lidar Signals

The actual signals used in this paper were all detected by a small Mie-scattering lidar developed at North Minzu University (106°06′ E, 38°29′ N). The system parameters of the Mie-scattering lidar are shown in Table 6. The data set used to train the network consisted of a total of 830 sets of data, including sunny, cloudy, cloudy, sandy, and other weather data, of which 90% were used for training and 10% were used for testing. Every collected original signal had a total of 10,000 data points, and the first 1000 data points were background noise, which was able to be used to calculate the average noise of the data acquisition system and was subtracted in the data preprocessing. Therefore, the starting point of the calculation was set to 1000.

Table 6. The system parameters of Mie scattering lidar.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Parameters Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laser</td>
<td>Nd:YAG laser</td>
</tr>
<tr>
<td>Wavelength</td>
<td>1064 nm, 532 nm, 355 nm</td>
</tr>
<tr>
<td>Single pulse energy</td>
<td>350 mJ@1064 nm, 170 mJ@532 nm, 80 mJ@355 nm</td>
</tr>
<tr>
<td>Impulse frequency</td>
<td>1–10 Hz</td>
</tr>
<tr>
<td>Pulse width</td>
<td>≤ 10 ns@1064 nm</td>
</tr>
<tr>
<td>Beam diameter</td>
<td>~ 9 mm@1064 nm</td>
</tr>
<tr>
<td>Divergence angle</td>
<td>≤ 0.5 mrad@1064 nm</td>
</tr>
<tr>
<td>Telescope</td>
<td>Schmidt–Cassegrain type</td>
</tr>
<tr>
<td>Diameter</td>
<td>300 mm</td>
</tr>
<tr>
<td>Field of view angle</td>
<td>0.4 mrad</td>
</tr>
<tr>
<td>Horizontal angle</td>
<td>0–360°</td>
</tr>
<tr>
<td>Pitching angle</td>
<td>0–90°</td>
</tr>
<tr>
<td>Detection range</td>
<td>100–13,000 m (Night), 100–10,000 m (Daytime)</td>
</tr>
</tbody>
</table>

Figure 9 shows the aerosol extinction coefficient profiles at a wavelength of 532 nm retrieved by the Klett method and the corresponding denoising effect. In Figure 9a, the original signal contains a lot of noise, and the useful signal is almost completely submerged in the noise above the height of 6 km. Figure 9b shows the aerosol extinction coefficient profile processed by the wavelet threshold method. Although the noise is filtered out, the signal details are seriously lost at the height of more than 8 km. In Figure 9c, the aerosol extinction coefficient profile is processed by the VMD method. Its noise reduction effect and signal detail retention are better than the wavelet threshold method, however, compared with the denoising method based on the CAENN method, the signal details are also lost to a certain extent. In Figure 9d, the aerosol extinction coefficient profile is processed by the CAENN method. Although the curve is rough, the details are well maintained. In order to quantitatively evaluate the denoising effect of these three methods, the SNR and MSE were
calculated, as listed in Table 7. It can be seen that the SNR of the method proposed in this paper was higher than the other two, while the MSE was lower than the other two, and the denoising effect was the best.

![Figure 9](image-url)

**Figure 9.** The aerosol extinction coefficient profiles obtained by the Klett method and its denoising effect at a wavelength of 532 nm. (a) the original signal, (b) the signal processed by the wavelet threshold method, (c) the signal processed by the VMD method, (d) the signal processed by CAENN method.

| Table 7. The SNR and MSE of lidar signals denoised by three methods shown in Figure 9. |
|-----------------|-----------------|-----------------|-----------------|
|                 | Wavelet Threshold | VMD             | CAENN           |
| SNR             | 19.8002         | 20.5140         | 21.4392         |
| MSE             | $6.4828 \times 10^{-7}$ | $6.1008 \times 10^{-7}$ | $5.68 \times 10^{-7}$ |

In order to further verify the denoising effect of the three methods for complex lidar signals, the extinction coefficient profiles containing cloud layer were employed for our goal. Figures 10 and 11 show the aerosol extinction coefficient profiles and denoising effect inverted by the Klett method. Compared with Figure 9, the two groups of signals have obvious cloud layers. In Figures 10a and 11a, the original signals contain a lot of noise. The useful signals are almost completely submerged in the noise above the height of 5 km. In Figure 10a, there is a cloud layer with a thickness of 1 km between 3 km and 4 km, and its extinction coefficient has reached 0.011 km$^{-1}$. However, in Figure 11a, a cloud layer with a thickness of 1 km appears at the height of 3 km to 4 km with an extinction coefficient of 0.049 km$^{-1}$; in addition, there is another cloud layer with a thickness of 2 km at the height of 4 km to 6 km with an extinction coefficient of 0.038 km$^{-1}$. In Figures 10b and 11b, the aerosol extinction coefficient profiles are processed by the wavelet threshold method.
Although the curves are very smooth and almost all the noise is filtered out, there are some significant distortions, especially at the peaks of the clouds, and the extinction coefficient value shows an obvious change compared with the original signals shown in Figures 10a and 11a. In Figures 10c and 11c, the aerosol extinction coefficient profiles are processed by the VDM method. The effect of noise reduction and signal detail retention are better than the wavelet threshold method. In Figures 10d and 11d, the aerosol extinction coefficient profiles are processed by the CAENN method. Although the curves are relatively rough, the details are maintained well. Moreover, this method does not achieve its denoising effect at the cost of the spatial resolution and distortion. Meanwhile, comparing VDM with the CAENN method, the signal details denoised by VDM method are lost to a certain extent. In order to evaluate further the processing effects of these three methods on the two groups of signals, the corresponding SNRs and MSEs were calculated respectively. Tables 8 and 9 list the SNR and MSE of signals denoised by three methods shown in Figures 10 and 11, respectively. It can be seen that the SNRs of the CAENN method proposed in this paper are higher than the other two, while the MSEs are lower than the other two, and the denoising effects are the best.

Figure 10. The aerosol extinction coefficient profiles containing a cloud layer obtained by the Klett method and its denoising effect at a wavelength of 532 nm. (a) the original signal, (b) the signal processed by the wavelet threshold method, (c) the signal processed by the VMD method, (d) the signal processed by CAENN method.
Figure 11. The aerosol extinction coefficient profiles containing two cloud layers obtained by the Klett method and its denoising effect at a wavelength of 532 nm. (a) the original signal, (b) the signal processed by the wavelet threshold method, (c) the signal processed by the VMD method, (d) the signal processed by CAENN method.

Table 8. The SNR and MSE of signals denoised by three methods shown in Figure 10.

<table>
<thead>
<tr>
<th>Method</th>
<th>Wavelet Threshold</th>
<th>VMD</th>
<th>CAENN</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>12.7456</td>
<td>14.1889</td>
<td>14.8419</td>
</tr>
<tr>
<td>MSE</td>
<td>$3.7665 \times 10^{-7}$</td>
<td>$2.6921 \times 10^{-7}$</td>
<td>$2.0179 \times 10^{-7}$</td>
</tr>
</tbody>
</table>

Table 9. The SNR and MSE of signals denoised by three methods shown in Figure 11.

<table>
<thead>
<tr>
<th>Method</th>
<th>Wavelet Threshold</th>
<th>VMD</th>
<th>CAENN</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>16.7403</td>
<td>17.9197</td>
<td>18.8352</td>
</tr>
<tr>
<td>MSE</td>
<td>$1.3339 \times 10^{-5}$</td>
<td>$9.0192 \times 10^{-6}$</td>
<td>$8.4257 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

4.4. Discuss

As seen in the simulation and measured data verification above, the denoising effect of the wavelet threshold method is the least effective among the three methods. The processed signal has significant distortion and the signal peak has a certain proportion of scaling. This denoising algorithm at the expense of signal fidelity is not desirable in practical applications. The VMD method is better than the wavelet threshold method in denoising effect and signal fidelity. However, the signal decomposition effect of VMD depends to a certain extent on the number of decomposition layers and the penalty factor, but these two parameters must be selected manually. It is usually difficult to obtain the optimal
combination of these two parameters, and the adaptive ability is poor. The algorithm proposed in this paper has strong adaptive ability, and its excellent denoising effect can be easily seen through the calculation of SNR and MSE. Because the number of signals with clouds in the training data set is small at present, the signal reconstructed by the model for the cloud signals contains a small amount of noise. If adding more data with clouds for training, the network learning ability and refactoring capability can be improved further, and the denoising effect will be improved well.

5. Conclusions

In this paper a method for denoising lidar return signals based on the CAENN method is proposed, which uses the encoding and decoding characteristics of the autoencoder to extract the deep features of lidar return signals layer by layer by constructing convolutional neural networks. Through the encoding–decoding process, the CAENN network can obtain the sparse expression of lidar return signals, which can convert the original lidar return signals with a lot of noise to lidar return signals with only effective signals, so as to filter out the noise. The encoding part extracts the features through the convolutional network, and eliminates the noise at the same time, while the decoding part completes the reconstruction of the data. To verify the feasibility of the method proposed, some simulations and experiments were carried out. The results show that the signal processed by CAENN method has the highest SNR ratio and the smallest MSE compared with the wavelet threshold and VMD methods, and its denoising effect is the most obvious. The algorithm proposed in this paper has strong adaptive ability, and its excellent denoising effect can be easily seen through the calculation of SNR and MSE. However, due to the scarcity of data involving special weather among the data sets, the processing of such data may lead to a small amount of noise remaining in the reconstructed signal. In future, we will collect more lidar signals in special weather to expand the data sets in order to train the network and optimize the model, thus further improving the universality of the model. This proposed method effectively improves the SNR of the lidar return signal, while retaining the complete characteristics of the signal, proving its effectiveness in denoising the lidar return signal.

Author Contributions: Data curation, M.H.; Funding acquisition, J.M.; Investigation, J.M.; Methodology, J.M.; Project administration, J.M.; Supervision, J.M.; Validation, J.L. and Y.Z.; Writing—original draft, M.H.; Writing—review & editing, J.M. and Q.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Nos. 61765001 and 61565001), the Natural Science Foundation of Ningxia Province (No. 2021AC02021), Plan for Leading Talents of the State Ethnic Affairs Commission of the People’s Republic of China, Innovation Team of Lidar Atmosphere Remote Sensing of Ningxia Province, the high level talent selection and training plan of North Minzu University, the Research Project of Serving Nine Key Industrial Projects for Ningxia of North Minzu University (No. FWNX20), and Ningxia First-Class Discipline and Scientific Research Projects (Electronic Science and Technology) (No. NXYLXK2017A07).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The aerosol particle size distribution and concentration data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this article.

References


