Title

EncoderDecoder Full Residual Deep Networks for Robust Regression and Spatiotemporal Estimation

Journal

IEEE Transactions on Neural Networks and Learning Systems, 32(9)

ISSN

2162-237X

Authors

Li, Lianfa
Fang, Ying
Wu, Jun
et al.

Publication Date

2021-09-01

DOI

10.1109/tnnls.2020.3017200

Peer reviewed

HHS Public Access

Author manuscript

IEEE Trans Neural Netw Learn Syst. Author manuscript; available in PMC 2022 September 01.

Published in final edited form as:

IEEE Trans Neural Netw Learn Syst. 2021 September ; 32(9): 4217–4230. doi:10.1109/TNNLS.2020.3017200.

Encoder-Decoder Full Residual Deep Networks for Robust Regression and Spatiotemporal Estimation

Lianfa Li,
State Key Lab of Resources and Environmental Information System, Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing, 100101 China

Ying Fang,
State Key Lab of Resources and Environmental Information System, Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing, 100101 China

Jun Wu,
Department of Environmental and Occupational Health, University of California Irvine, Irvine, CA, 92697, USA

Jinfeng Wang,
State Key Lab of Resources and Environmental Information System, Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing, 100101 China

Yong Ge
State Key Lab of Resources and Environmental Information System, Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing, 100101 China

Abstract

Although increasing hidden layers can improve the ability of a neural network in modeling complex non-linear relationships, deep layers may result in degradation of accuracy due to the problem of vanishing gradient. Accuracy degradation limits the applications of deep neural networks to predict continuous variables with a small sample size and/or weak or little invariance to translations. Inspired by residual convolutional neural network in computer vision, we developed an encoder-decoder full residual deep network to robustly regress and predict complex spatiotemporal variables. We embedded full shortcuts from each encoding layer to its corresponding decoding layer in a systematic encoder-decoder architecture for efficient residual mapping and error signal propagation. We demonstrated, theoretically and experimentally, that the proposed network structure with full residual connections can successfully boost the back-propagation of signals and improve learning outcomes. This novel method has been extensively evaluated and compared with four commonly-used methods (i.e., plain neural network, cascaded residual autoencoder, generalized additive model and XGBoost) across different testing cases for

(Corresponding author: Lianfa Li.) lilf@lreis.ac.cn.

continuous variable predictions. For model evaluation, we focused on spatiotemporal imputation of satellite aerosol optical depth with massive non-randomness missingness, and spatiotemporal estimation of atmospheric fine particulate matter ≤ 2.5 μm ($PM_{2.5}$). Compared to the other approaches, our method achieved the state-of-the-art accuracy, had less bias in predicting extreme values, and generated more realistic spatial surfaces. This encoder-decoder full residual deep network can be an efficient and powerful tool in a variety of applications that involve complex non-linear relationships of continuous variables, varying sample sizes, and spatiotemporal data with weak or little invariance to translation.

Keywords

Deep learning; encoder-decoder full residual deep network; non-linear regression; spatiotemporal modeling; prediction of satellite AOD and $PM_{2.5}$, bias

I. Introduction

DEEP learning has achieved great successes in various domains including bioinformatics, material science, reinforcement learning, computer vision, natural language processing and others [1] due to a series of breakthroughs in back-propagation [2], fast graphics processing units [3], activation functions such as rectified linear unit (ReLU) [4], convolutional neural network (CNN) [5], long short-term memory [6], generative adversarial network [7], and deep belief network [8] etc.

One crucial aspect of deep learning is network depth [9]. Deep networks have more trainable parameters (e.g., weights and bias) to capture complex relationships among variables and much better generalization than shallow ones [10]. The earlier obstacle of vanishing or exploding gradient caused by deep hidden layers has been mostly addressed by using efficient activation functions such as ReLU, normalization initialization and batch normalization, given sufficient training samples. Whereas activation and normalization partially solve the convergence issue in deep networks, too many hidden layers may quickly saturate or degrade accuracy, as shown in many experiments of CNN [10], [11] and multilayer perceptron (MLP). Further, deep neural networks usually need large training samples to find an optimal solution as small samples often result in non-convergence or degraded accuracy.

Residual connections have been used in CNN to boost learning efficiency and address the issue of accuracy degradation, with a wide range of applications including classification [12], segmentation [13], image super-resolution [14], [15] and compression [16], and crowd flows predictions [17] etc. Autoencoder and residual learning have been combined in two recent studies for imputation of missing modalities [18] and image restoration [19]. In these two studies, Tran et al. [18] developed cascaded residual autoencoders (CRA) with one hidden layer in each shallow autoencoder; Zini et al. [19] applied a residual dense block in the latent coding layer of an autoencoder. In the CRA, each autoenocoder's input was the summation of the input and output of the previous autoencoder, and its desired output was the difference between the input (i.e. incomplete) data sample and the complete data sample, as the output of residual. Thus, each autoencoder was trained independently and

subsequently joint optimization was applied for the CRA [18]. This layer-by-layer residual learning in the CRA is different from the end-to-end learning used in Zini's method in which residual units (Supplementary Fig. S1) were directly embedded in the deep neural networks [12], [18], [19].

CNN, a data-intensive learner, is powerful in handling data that are highly invariant to translation (e.g., scale, rotation, shift and position; mostly images or videos), but challenges exist for CNN to handle small samples and data with weak or little invariance to translation. One example of such data is satellite-based aerosol optical depth (AOD) [20], which has weak translation invariance in space as it is affected by multiple physical factors such as meteorology, emission sources and elevation, and the complex atmospheric chemical processes involving these factors [21]. Another example is the highly heterogeneous spatiotemporal distributions of environmental pollutants, which likely have weak translation invariance in space due to the influence of multiple physical and chemical factors, and their complex interplay. In addition, measurements of pollutant concentrations are generally small in sample size due to sparsely-located monitoring stations [22]. For the two examples above with weak translation invariance and/or a small sample size, CNN may be not able to account for the complex and likely non-linear influence of various physical parameters on the target variables (i.e. AOD and pollutant concentrations). Deep MLP, a class of feed-forward neural network consisting of an input layer, multiple hidden layers and an output layer, may be effective in modeling complex non-linear relationships due to the use of multiple layers, the non-linear activation, and the flexible network structure [23]. However, with increased hidden layers the MLP also faces the challenge of vanishing gradient and degradation of accuracy, particularly for a small sample size. Residual learning [11], which has been extensively used in CNN, can be applied in deep MLP to enhance learning, although few studies [18] have reported the use of residual learning in deep MLP.

In this paper, we present a new architecture of encoder-decoder full residual deep network as a robust solution of deep learning, particularly for applications in regression and spatiotemporal prediction of continuous variables. This method is broadly inspired by residual convolutional neural network in computer vision and recent findings in neuroscience on crucial shortcuts in animals' brains [23]. We introduce full residual connections into the encoder-decoder architecture. Different from residual connections stacked continuously in ResNet [24], we take advantage of the symmetrical structure of encoding and decoding layers in the architecture, and leverage the shortcuts of identify mapping as residual connections from the encoding layers to their corresponding decoding layers. Different from the CRA with cascaded shallow autoencoders [18], our full residual connections are embedded end-to-end in a deep encoder-decoder structure for more efficient signal propagation and learning. Further, different from the use of residual blocks only in the latent layer [19], our residual shortcuts are fully connected from each encoding layer (including the input layer) to its corresponding decoding layer in a nested way. Thus, forward and backward error signals can be propagated directly and fully between the encoding and decoding layers. Additionally, we use non-linear ReLU or Exponential Linear Unit (ELU), and/or linear activation to ensure the optimal property of efficient backpropagation of errors within the full residual deep network. Our proposed architecture can be fully implemented in both CNN and MLP. In this paper, we mainly focus on residual

deep MLP and illustrate its applications in multiple case studies with either small samples or weak translation invariance in data. We test multiple datasets, focusing on imputation of massive non-random missing data in satellite AOD and prediction of atmospheric fine particulate matter ≤ 2.5 μm ($PM_{2.5}$) at a high spatiotemporal resolution. The proposed full residual deep network has been extensively evaluated, and compared to commonly-used methods.

This paper has three main contributions to the literature: 1) we propose a novel full residual deep network embedded in the symmetrical encoder-decoder structure that can considerably boost the learning for regression and spatiotemporal estimation of continuous variables; 2) we show theoretically and experimentally that full residual connections in the proposed network structure can achieve more efficient error signal propagation, more efficient learning, and more accurate prediction than no or limited residual connections; 3) we demonstrate the state-of-the-art performance of the proposed method in spatiotemporal variable imputation and prediction, for data with both small and big sample sizes, and for data with weak or little invariance to translation.

II. Related Work

A. Encoder-Decoder

Encoder-decoder is a type of neural network architecture with a possibly symmetrical structure (Fig. 1) from the encoding layers to the decoding layers with the input and output layers, and the middle layer of latent representation [10]. The motivation of this architecture is to efficiently learn feature representations (latent representation) from raw inputs using an encoder module and take this feature representation to generate an output using a decoder module. If the input and output are the same, this architecture degenerates to an autoencoder [25], [26]. Encoder-decoder, a general architecture of deep learning for automatic and efficient learning of representation, is used in U-Net [27], SegNet [28], Seq2Seq [29], and other applications [10].

Assuming a d-dimension input, $\mathbf{x}$, an m-dimension output, $\mathbf{y}$, weight matrix, $\mathbf{W}$, bias vector, $\mathbf{b}$ the set of parameters, $\theta_{\mathbf{w},\mathbf{b}}$, the layer index, L, we have the following mapping formula:

$$\theta_{\mathbf{W},\mathbf{b}}(\mathbf{x}):\mathbf{R}^d \rightarrow \mathbf{R}^m \tag{1}$$

The parameters $\theta_{\mathbf{w},\mathbf{b}}$ can be obtained by minimizing the loss function between the ground truth ($\mathbf{y}$) and the predictions ($\mathbf{y}'$) over the training data.

The U-shape encoder-decoder provides a symmetrical topology to implement the functionality to learn an efficient latent data representation. We can embed multiple hidden layers in encoding with sequentially decreased number of nodes to compress high dimensional data into powerful latent representations in the coding layer which, with less noise than original data, is beneficial for training and generalization [30].

B. Residual Learning

Artificial neural network, inspired by the biological neural networks that constitutes animal brains [31], is crucial in deep learning. Recent findings show the importance of shortcuts in the brains for coordinated motor behavior and reward learning [23]. Such shortcuts collaborate with plain connections to accomplish complex functionality. Although the mechanism about shortcuts in brains is unclear, similar ideas of skip connections or shortcuts have been used in artificial neural network. Studies show that residual vectors are powerful shallow representation in image recognition [32]. Consequently, residual shortcuts in CNN have been proposed to tackle the issue of accuracy degradation in learning [12], [24].

In a typical residual CNN, each residual unit (Supplementary Fig. S1) includes two or three continuous convolutional layers with optional batch normalization and ReLU activation function, and a residual shortcut connection between the input of the first layer and the output of the last layer; all of the residual units are stacked continuously to increase the depth and the generalization of the model (Supplementary Fig. S2, based on [12]). In this residual CNN, the shortcut of identity mapping is employed in a continuously stacked sequence (similar to ensembles of relatively shallow networks [33]) to implement residual connections. Residual CNN has been extensively applied in many domains, including computer vision [14], [18], [19] or remote sensing [34] that typically involve intensive training samples with invariance to translations.

C. Spatiotemporal Modeling

Due to its capability of capturing neighborhood information in a spatiotemporal domain, CNN has been used for spatiotemporal modeling, e.g., action segmentation [35] and understanding [36] of video data. From 2017, residual learning has been introduced into CNN for spatiotemporal modeling, first for prediction of citywide crowd flows [17], then vehicle counting [37], [38], and prediction of influenza trends [39] etc. In these spatiotemporal CNN applications, residual units were used to improve learning efficiency. More recently, residual CNN has been combined with long short-term memory for passenger flows predictions [40].

Although residual CNN has been increasingly used in spatiotemporal modeling for powerful learning ability, such applications are data-driven and require minimal missing values in the input data. Since most air pollution data come from sparsely-distributed monitoring locations, CNN may not work well to predict pollutant concentrations at a high spatiotemporal resolution under the conditions of limited training data with a coarse spatiotemporal resolution or incomplete predictors with substantial missing data.

In this paper, we focus on two case applications, spatiotemporal imputation of massive missing values of satellite AOD and prediction of spatiotemporal $PM_{2.5}$ at a high resolution. For AOD imputation, typical methods include spatial interpolation [41], [42], forward-forward neural network (plain MLP) [43], image retrieval algorithms [44], [45], and non-linear generalized additive models (GAM) [46] etc. These methods achieved small validation R-squared (R^2) ranging from 0.18 to 0.44 when comparing to the AErosol

RObotic NETwork (AERONET) AOD [47]. For spatiotemporal $PM_{2.5}$ prediction at a high resolution, typical methods include geographically and temporally weighted regression [48], [49], mixed-effect model [50], [51], two-stage models [52], GAM [46], hybrid neural network [43], random forest [53], and XGBoost [54] etc. These methods achieved validation R^2 ranging from 0.57 to 0.87. The previous models in the literature have various limitations, including inconvenience of use (e.g. two stage and hybrid models), loss of information using decision trees (e.g. random forest and XGBoost), and simple model structure that limits its capability of handling complex non-linear relationships among multiple variables (e.g. GAM). Plain MLP has been used in $PM_{2.5}$ estimation recently, with only moderate performance [43], [55]. A full residual deep network is expected to better capture complex non-linear relationships between the spatially and/or spatiotemporal varying predictors and the target variables.

III. Full Residual Deep Network

A. Encoder-Decoder Architecture

For the encoding layers of our architecture, every hidden layer has a different number of nodes, which can introduce variations in extracting informative latent representation potentially beneficial for effective learning. Based on the core of an encoder-decoder, residual connections are introduced through the construction of full skip connections or shortcuts to jump over the layers between the encoding layers and their corresponding decoding layers. This residual network, as a type of special neural network, can preserve the information in the input or earlier layers and reduce vanishing gradient and degradation of accuracy in deep networks [56]. A U-shape symmetrical encoder-decoder is a natural option for such a residual network given that a residual connection requests the same number of nodes for the two layers involved (a shallow layer and its corresponding deep layer). Fig. 2 presents the architecture of a typical encoder-decoder full residual network that includes the layers of an input and an output, symmetrical k+1 encoding (1 input layer and k hidden layers) and k+1 decoding hidden layers, and a middle latent representation layer. We can add activation functions and batch normalization to each layer if necessary.

This architecture is applicable to both MLP and CNN. For MLP, the number of nodes for each layer, $n_i (i = 1, \ldots, k + 1)$, is given to construct the encoding and the decoding layers, and subsequently implement residual connections. For CNN, encoding can be implemented using downsampling like a pooling layer to obtain the latent representation layer, and decoding can be implemented using upsampling to obtain the target output. Unlike U-Net [27] that uses feature concatenations to implement shortcuts in its U-shape encoder-decoder structure [57], we used vector additions to implement residual mapping of full shortcuts in our architecture. Thus, our architecture is a residual network with short and long shortcuts from the encoder to the decoder. Compared with feature concatenations in U-Net, vector additions of identity mapping do not increase the number of parameters and model complexity. The input of an encoding layer can be added to the output of its corresponding decoding layer to implement residual mapping. In this paper, we refined and applied a full residual deep MLP to spatiotemporal modeling of satellite AOD imputation and $PM_{2.5}$ prediction.

For residual deep MLP, there are two options for t target variables of $\mathbf{y}$ to be output:

1) Output of Target Variables: The target variables can be treated as an independent output layer in the encoder-decoder architecture (option 1 in Fig. 2). We can define the total loss function, L as the following:

$$L(\theta_{\mathbf{W},\mathbf{b}}) = \frac{1}{N}\ell_O\left(\mathbf{y}, f_{\theta_{\mathbf{W},\mathbf{b}}}(\mathbf{x})\right) + \Omega(\theta_{\mathbf{W},\mathbf{b}}) \tag{2}$$

where $\ell_O\left(\mathbf{y}, f_{\theta_{\mathbf{W},\mathbf{b}}}(\mathbf{x})\right)$ represents the loss function of mean square error (MSE) for regression or cross entropy for classification,$\mathbf{y}$ is the observed value, $f_{\mathbf{W},\mathbf{b}}(\mathbf{x})$ is the predicted value, $\theta_{\mathbf{W},\mathbf{b}}$ represents the network parameters, $\mathbf{W}$ and $\mathbf{b}$ are to be optimized, and $\Omega(\theta_{\mathbf{W},\mathbf{b}})$ denotes the regularization for $\theta_{\mathbf{W},\mathbf{b}}$ (L1, L2 or elastic net [58]).

2) Output of Explanatory Variables and Target Variables: The input explanatory variables and the target variables can be used as the output layer (option 2 in Fig. 2). This option enables more sharing of the parameters among the explanatory and target variables, and more constraints on the target variables. The loss function can be defined as:

$$\begin{aligned} L(\theta_{\mathbf{W},\mathbf{b}}) = \frac{1}{N}\Big[&\ell_O\left(\mathbf{y}, f_{\theta_{\mathbf{W},\mathbf{b}}}(\mathbf{x})\right) \\ &+\ell_{MSE}\left(\mathbf{x}, f_{\theta_{\mathbf{W},\mathbf{b}}}(\mathbf{x})\right)\Big] + \Omega(\theta_{\mathbf{W},\mathbf{b}}) \end{aligned} \tag{3}$$

where $\ell_{MSE}\left(\mathbf{x}, f_{\theta_{\mathbf{W},\mathbf{b}}}(\mathbf{x})\right)$ denotes the MSE loss function for the output of the input, $\mathbf{x}$.

Options 1 and 2 differ in the placement of the target variables within the network. Comparison of Eq. 2 and 3 shows one constraint on the parameters in terms of prediction of $\mathbf{y}$ in option 2. This constraint works as regularizers for $\mathbf{y}$ [10]. When sufficient samples are available, option 2 can effectively prevent over-fitting. When the sample size is limited, additional regularizers in option 2 may increase training errors, thus option 1 is preferred.

B. Residual Connections and Implementation

Skip connections or shortcuts have been added in neural networks to address the issues of vanishing or exploding gradients [11], [59] and degradation of accuracy in residual CNN [12], [24]. We use the shortcut connection of identity mapping from each encoding layer to its corresponding decoding layer to implement nested residual connections from the outermost layers to the innermost layers (Fig. 2). Depending on how activation and batch normalization are implemented after the outputs of the hidden layers, three options are available for the output of residual connection: none added, only activation added, both activation and batch normalization added (Supplementary Fig. S3).

Theoretically, we show below that residual connections can effectively improve information backpropagation in learning of the encoder-decoder full residual network. Assuming the middle latent layer, M, the decoding layer, l, and its mirror decoding layer, L, with addition of residual identity connection, we have:

$$\mathbf{y}_L = \mathbf{x}_l + f_L(\mathbf{x}_L, \mathbf{W}_L) \tag{4}$$

where $\mathbf{x}_l$ and $\mathbf{y}_l$ are input and output of the encoding layer, l, respectively, $\mathbf{x}_L$ and $\mathbf{y}_L$ are input and output of the decoding layer, L, respectively, $\mathbf{W}_l$ is the parameters (including the bias) for the l layer, and $f_L(\mathbf{x}_L, \mathbf{W}_L)$ is the sequence of the weighted summation of the L layer input, $\mathbf{x}_L$ and activation.

Since L is a deeper layer for l with residual mapping between both layers, we can rewrite Eq. 4 as:

$$\mathbf{y}_L = \mathbf{x}_l + f_L(g_L(f_l(\mathbf{x}_l, \mathbf{W}_l)), \mathbf{W}_L) \tag{5}$$

where $g_L(f_l(\mathbf{x}_l, \mathbf{W}_l))$ denotes the multi-layer function for $\mathbf{x}_L$ with $\mathbf{x}_l$ as input, where $\mathbf{x}_L = g_L(f_l(\mathbf{x}_l, \mathbf{W}_l)) = f_{L-1}(\ldots f_l(\mathbf{x}_l, \mathbf{W}_l)\ldots, \mathbf{W}_{L-1})$.

According to automatic differentiation [60], we can obtain the general derivative of the loss function, L for $\mathbf{x}_l$ that is used to compute the gradients for the parameters, $\mathbf{W}_{l-1}$:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{x}_l} &= \frac{\partial L}{\partial f'_L(\mathbf{y}_L)} \frac{\partial f'_L(\mathbf{y}_L)}{\partial \mathbf{y}_L} \frac{\partial \mathbf{y}_L}{\partial \mathbf{x}_l} \\ &= \frac{\partial L}{\partial f'_L(\mathbf{y}_L)} \frac{\partial f'_L(\mathbf{y}_L)}{\partial \mathbf{y}_L} \\ &\left(1 + \frac{\partial}{\partial \mathbf{x}_l}(f_L(g_L(f_l(\mathbf{x}_l, \mathbf{W}_l)), \mathbf{W}_L))\right) \end{aligned} \tag{6}$$

where $f'_L(\mathbf{y}_L)$ is the possible activation function or batch normalization for the output of the L layer.

If we use the residual connection of option 1 in Supplementary Fig. S3-a (no activation and batch normalization after addition of the shortcut identity connection), we can get a simple version of Eq. 6:

$$\frac{\partial L}{\partial \mathbf{x}_l} = \frac{\partial L}{\mathbf{y}_L} \cdot \left(1 + \frac{\partial}{\partial \mathbf{x}_l}(f_L(g_L(f_l(\mathbf{x}_l, \mathbf{W}_l)), \mathbf{W}_L))\right) \tag{7}$$

There is one constant term, 1 in $\frac{\partial \mathbf{y}_L}{\partial \mathbf{x}_l}$ of Eq. 6 and 7 that makes the information of $\frac{\partial L}{\mathbf{y}L}$ directly propagated to the early layer, $\mathbf{x}_l$ without addition of any weight layers. Further, $\frac{\partial}{\partial \mathbf{x}_l}(f_L(g_L(f_l(\mathbf{x}_l, \mathbf{W}_l)), \mathbf{W}_L))$ is not always equal to −1 to cancel out the gradient, $\frac{\partial L}{\partial \mathbf{x}_l}$ for mini-batch learning. This property can reduce gradient vanishing during backpropagation and subsequent degradation of accuracy. Thus, such shortcut connections can improve the training of networks in collaboration with the plain connections in deep layers.

For option 2 and 3 in Supplementary Fig. S3, the activation function of ReLU, ELU or linear unit, and/or batch normalization can be added [$f'_L(\mathbf{y}_L)$ in Eq. 6] to better model non-linear

relationships and maintain the nice property of direct backpropagation of error signals from the deep layers to the early layers (aforementioned according to Eq. 6).

Based on the architecture, we adopt the nested shortcuts of identity mapping from the outermost layers to the innermost layers for residual learning (Fig. 2), which is different from the short shortcuts (commonly jumping over two or three layers) stacked continuously in residual CNN (ResNet) [12]. In the nested structure, in addition to plain backpropagation, error information is directly transferred in the outmost layers of shortcut connections (from the last layer to the first layer). Then, such backpropagation occurs from the second nested layers till the innermost layers. This nested structure has advantages over the residual CNN where error information is backpropagated along a longer path of multiple stacked residual units.

For the proposed network structure (Fig. 2), an inner residual connection may recursively affect the outer residual connections by backpropagation. To illustrate this, Fig. 3 shows two simplified cases with one residual connection vs. two connections surrounding the middle latent layer.

For the innermost residual connection (Fig. 2-b) from l_k to L_k, we have:

$$\begin{aligned}\mathbf{y}_{L_k} &= f_{L_k}(\mathbf{y}_M) + \mathbf{x}_{l_k} = f_{L_k}\left(f_M\left(\mathbf{y}_{l_k}\right)\right) + \mathbf{x}_{l_k} \\ &= f_{L_k}\left(f_M\left(f_{l_k}\left(\mathbf{x}_{l_k}\right)\right)\right) + \mathbf{x}_{l_k}\end{aligned} \tag{8}$$

where M represents the middle latent layer, and $f_i(\mathbf{x})$ is a sequence of the weighted summation for the i layer's input, $\mathbf{x}$, and activation (e.g., ReLU). The weights and biases are omitted for simplified notations.

For the outer residual connection from l_{k-1} to L_{k-1}, we have:

$$\mathbf{y}_{L_{k-1}} = f_{L_{k-1}}\left(\mathbf{y}_{L_k}\right) + \mathbf{x}_{l_{k-1}} \tag{9}$$

For Fig. 3-b where the innermost residual connection is available, substituting Eq. 8 into Eq. 9, we get:

$$\mathbf{y}_{L_{k-1}} = f_{L_{k-1}}\left(f_{L_k}\left(f_M\left(f_{l_k}\left(\mathbf{x}_{l_k}\right)\right)\right) + \mathbf{x}_{l_k}\right) + \mathbf{x}_{l_{k-1}} \tag{10}$$

where $\mathbf{x}_{l_k} = f_{l_{k-1}}\left(\mathbf{x}_{l_{k-1}}\right)$.

Then, we can get the derivative of the loss function, L for $\mathbf{x}_{l_{k-1}}$:

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{x}_{l_k-1}} &= \frac{\partial L}{\partial \mathbf{y}_{L_k-1}}\left(\frac{\partial f_{L_k-1}(\mathbf{x}_{L_k-1})}{\partial \mathbf{x}_{L_k-1}}\frac{\partial \mathbf{x}_{L_k-1}}{\partial \mathbf{x}_{k-1}}+1\right)\\ &= \frac{\partial L}{\partial \mathbf{y}_{L_k-1}}\left(\frac{\partial f_{L_k-1}(\mathbf{x}_{L_k-1})}{\partial \mathbf{x}_{L_k-1}}\right)\\ &\left(\frac{\partial f_{L_k}(\mathbf{x}_{L_k})}{\partial \mathbf{x}_{l_k-1}}+\frac{\partial f_{l_k-1}(\mathbf{x}_{l_k-1})}{\partial \mathbf{x}_{l_k-1}}\right)+1\end{aligned} \tag{11}$$

In Eq. 11, the term of $\partial f_{l_k-1}(\mathbf{x}_{l_k-1})/\partial \mathbf{x}_{l_k-1}$ is the single-layer derivative of the l_k-1's output for its input; intuitively, besides the constant term of 1, this term can complement the (vanishing) gradient for $\mathbf{x}_{l_k-1}$ since it has a shorter path than $\partial f_{L_k}(\mathbf{x}_{L_k})/\partial \mathbf{x}_{l_k-1}$. For mini-batch learning, $\partial f_{L_k}(\mathbf{x}_{L_k})/\partial \mathbf{x}_{l_k-1}$ does not always cancel out $\partial f_{l_k-1}(\mathbf{x}_{l_k-1})/\partial \mathbf{x}_{l_k-1}$. Thus, for the outer residual connections, error backpropagation may be further improved by the inner residual connection if linear or semi-linear activation function (e.g., ReLU) is used. Each residual connection enables direct backpropagation of errors from its decoding layer to its encoding layer and can recursively enhance information backpropagation for the outer layers. Lack of shallow-to-deep layer residual connections can hinder backpropagation due to the presence of multiple layers between the shallow encoding layer and the deep decoding layer, potentially resulting in multiplication of small derivatives and vanishing gradient. Theoretically, our method has an optimal network structure with all residual connections from the encoding layers to the decoding counterparts, which allows for highly efficient information backpropagation. Compared to short residual units in ResNet [12], [24], residual shortcuts in our architecture can be long, jumping over more layers to implement identity mapping, which is theoretically shown above to improve efficiency of error back-propagation by directly informing the deep layers with low-level information from the early layers. In addition, an inner residual connection is theoretically shown to boost back-propagation in the outer residual connections recursively. To differentiate our proposed method from the other existing approaches, we name our method "encoder-decoder full residual deep network".

We developed an iterative version of the proposed full residual deep network (Algorithm 1) for option 3 in Supplementary Fig. S3. In this algorithm, BN denotes

Algorithm 1:

Full Residual Regression Deep Network

Input : Number of features, n_f; list of the numbers of nodes for each layer, ln ; number of target variables, k; list of activation functions for each layer, la; dropout rate, d_r.
Output: Model of full residual deep network.

Generate the input layer, $layer_{in}$ according to n_f ;
Set $tlayer$=$layer_{in}$;
for *each* i,_ *in enumerate*(ln) **do**
  Add a fully-linked layer ($ln[i]$ nodes) to $tlayer$;
  if $i <(length(ln) - 1)$ **then**
    Push $tlayer$ to the stack, S ;
  else
    Add a dropout layer (rate=d_r) to $tlayer$;
  Add ACT ($la[i]$) or/and BN to $tlayer$;
end for
for *each* i,_ *in reversed(enumerate*(ln)) **do**
  Pop p_{layer} from the stack, S ;
  Add a fully-linked layer ($ln[i]$ nodes) to $tlayer$;
  Add ACT ($la[i]$) or/and BN to $tlayer$;
  Add the addition of two layers: $tlayer$ + p_{layer} to $tlayer$;
  Add ACT ($la[i]$) or/and BN to tlayer ;
end for
Add the addition of two layers: $tlayer$ + $layer_{in}$ to $tlayer$;
Add ACT or/and BN to $tlayer$;
Add the output layer with k nodes (option 1) or $k + m$ nodes (option 2) to $tlayer$;
Return the model with input ($layer_{in}$) and output ($tlayer$).

batch normalization and ACT denotes activation function; a stack is used to store the early encoding layers and then pop them sequentially to construct the residual connections for the popped layers and their corresponding deep layers. We published this algorithm with

partial test data on the Github (https://github.com/lspatial/resautonet), and in the resautonet package of both Python (https://pypi.org/project/resautonet/) and R Statistics (https://cran.r-project.org/web/packages/resautonet/).

In the proposed network (Algorithm 1 and Fig. 2), the basic building block consists of a shallow layer and its corresponding deep counterpart with activation and batch normalization. This building block with optimal choice of its components is crucial in our method. For activation, we choose the efficient activation function of ReLU or ELU for most layers except the output layer. ReLU and ELU have identity function and the constant derivative of 1 for x≥0. Thus, they can partially keep efficient backpropagation. For x≤0, ReLU is 0 but ELU has an exponential function ($\alpha(e^x - 1)$) with exponential derivatives. With similar property for the positive input as ReLU, ELU can well capture non-linear characteristics for negative input [61]. For the output layer, we choose tanh or linear activation function. The tanh function can better capture non-linearity than logistic activation with its symmetrical range around the mean of 0 [62]. Batch normalization can be added to each hidden layer to solve internal covariate shift [63] and to speed up the learning process.

C. Training and Predicting

Our network architecture is flexible and can accommodate different types of variables (e.g. lagged and non-lagged) to model complex associations in the general applications of spatiotemporal regression. The lagged variables have the advantage of capturing autocorrelation in spatiotemporal prediction [64]. Given different scales of the input (x) and the output (y) variables, normalization (e.g., standardization) is required for both. Network training (Fig. 4-a) aims to optimize the following objective function:

$$\theta^{opt}_{\mathbf{W},\mathbf{b}} = \underset{\theta_{\mathbf{W},\mathbf{b}}}{\operatorname{argmin}} L\left(f_{\theta_{\mathbf{W},\mathbf{b}}}(\mathbf{x}), \mathbf{y}\right) \tag{12}$$

where $\theta^{opt}_{\mathbf{W},\mathbf{b}}$ denotes an optimal solution for the network parameters, and the total loss function, L, is given in Eq. 2 or 3, depending on selection of the output option (Fig. 2). We used Adam [65] as the optimizer. Sensitivity analysis was conducted to find an optimal structure (the number of encoding layers and the number of nodes for each layer). Grid search was conducted to obtain the optimal hyper-parameters including initial learning rate, mini-batch size and dropout rate etc. He normalization [66] was used to initialize the parameters.

After the optimal model is obtained, it can be used to make predictions (Fig. 4-b). The new input data for model predictions, once normalized, are fed to the trained model to generate outputs that are further inversely normalized to the original scale of the target variables. Although spatially and/or temporally lagged variables can be used to capture autocorrelation in spatiotemporal prediction, the use of these lagged variables requires continuous data in space and/or time, which is not feasible in model predictions where the target points of prediction are not uniformly distributed in space and/or time. This is the case for $PM_{2.5}$ prediction in environmental health studies since locations of interest (e.g. residential homes, work places) vary greatly in space. Similarly, lagged variables were not used in AOD

imputation due to the large percent and irregular missing AOD data in space and time. We used spatially varying variables such as coordinate and their derivatives to capture spatial correlation, and used multi-scale temporal variables (day of year and month index) to capture multi-scale temporal correlation. In addition, spatiotemporally varying meteorology variables were used to capture spatiotemporal variability of the target variable.

IV. Experiments

A. Test Datasets

To evaluate the performance and generalization of the proposed method, we tested a simulated dataset with a small sample size and six benchmark datasets from the publicly available UCI repository of machine learning (three for classification and three for regression) (http://archive.ics.uci.edu/ml). Then, our approach was applied for 1) spatiotemporal imputation of massive non-random missingness of satellite AOD (over 50% missing values), and 2) spatiotemporal estimation of $PM_{2.5}$. The main target variables for the two applications were daily Multiangle Implementation of Atmospheric Correction Aerosol Optical Depth (MAIAC) AOD in 2015 (365 days) and daily ground $PM_{2.5}$ concentrations in 2015 for the Beijing-Tianjin-Tangshan metropolitan area, China (Supplementary Fig. S4), respectively. Independent tests were conducted using AERONET AOD and the measured $PM_{2.5}$ from the US Embassy monitoring site in Beijing, as the ground truth.

Nine datasets from four case studies were used in model evaluation (Table I). For AOD imputation and $PM_{2.5}$ prediction, we used stratified sampling to ensure even distribution of training and test samples across space and time. For each dataset, we drew 20% of data for independent test, 20% from the rest of 80% samples (16% in total) for validation, and the rest of 64% samples for model training. Please see Supplementary Section I for details about these datasets and tests.

B. Investigation on the Residual Connections in Training

We examined the influence of residual connections on model training by conducting benchmark comparisons between the full residual deep network and the deep plain network (i.e. a deep MLP without residual connections). Both networks were based on an encoder-decoder structure for fair comparison. Our results (Supplementary Table SI) show that the full residual deep network consistently and in most cases considerably outperformed the deep plain network. More detailed results for the full residual deep network follow: (1) the simulated dataset: a 25% increase in R^2 and 38% decrease in root mean square error (RMSE) in an independent test; (2) three classification dataset from the UCI repository: an increase of 1–4% in accuracy, and an increase of 0–3% in area under receiver operating characteristic curve, typically used in classification in the independent test; (3) three regression datasets from the UCI repository: an increase of 3–58% in R^2 and a decrease of 0.04–7.98 in RMSE in the independent test; (4) MAIMC AOD: an average increase of 7% in R^2 and an average decrease of 0.01 in RMSE in the independent test (Supplementary Fig. S5 showing distributions of R^2 and RMSE of different models across 365 days of 2015); (5) $PM_{2.5}$: an increase of 20% in R^2 and a decrease of 14.46 $\mu g/m^3$ in RMSE in the independent test (Fig. 5 for the scatter plots of the observed vs. predicted values).

The training curves of the loss and performance metrics (R^2 for regression; accuracy for classification) are shown in Supplementary Fig. S6. For MAIAC AOD, we present a typical day; the other days had similar trends. Overall, the full deep residual network performed better (lower loss and higher R^2) than the deep plain network. Further, the residual deep network converged quickly, illustrating its high learning efficiency. For MAIAC AOD imputation, the non-residual plain networks did not converge in 28 out of 365 days, while the deep residual network converged on all 365 days. The scatter plots of the simulated vs. predicted values in the independent tests are presented for the simulated dataset (Supplementary Fig. S7) and $PM_{2.5}$ estimation (Fig. 5), showing that the full residual deep network had less overestimation at low values and much less underestimation at high values than the non-residual plain network.

In addition, we investigated the influence of different residual connections (the number of residual connections and their placements in the network) on model performance through independence tests on the simulated data, AOD imputation and $PM_{2.5}$ prediction. Specifically, we examined the number of residual connections ranging from zero to the maximum value (5 for the simulated data; 6 for AOD and $PM_{2.5}$), and all the combinations (each combination called a scenario) of different placements of the residual connections in the network for each number of connections. The maximum number of connections was determined empirically depending on sample size and complexity of the problem. For the network with only one residual connection, Fig. 6 shows different placements (from the innermost connection to the outermost one) in the network. In order to reduce the uncertainty in initialization and local optimization, we trained the network of each scenario 100 times and summarized the statistics (means and boxplots) of test performance metrics (R^2 and RMSE) from the 100 trained models. The results (Table II) differed by the location of the single residual connection in the network. For AOD, only two days of results are shown here due to space limitation. The residual connection in the two mid-layers or the outermost layers seemed to perform better than that in the innermost layers. For the networks with more than one residual connection, the results showed possibly substantially differences in test R^2 and RMSE between different placements for the same number of residual connections.

Further, the increase in the number of residual connections generally progressively improved the performance of the deep network (Table III; Fig. 7 for the boxplot of test RMSE, and Supplementary Fig. S8 for the boxplot of test R^2). This improvement was pronounced for the simulated data and the $PM_{2.5}$ data that had a small sample size. Fig. 8 shows the learning curves for different numbers (ranging from 0 to 6) of residual connections for $PM_{2.5}$ prediction; the other datasets had similar patterns (results not shown). We observed better performance for the models with more residual connections, and the best performance for the full connection model. These results consistently support the use of full residual connections as the optimal model structure, which has been analyzed theoretically in Section III-B.

When the network scale (indicated by the number of hidden layers) increases, the full residual connections may likely improve model performance with less degradation of

accuracy than the deep plain network (Table IV), indicating the robustness of our model to the change in network scales.

C. Imputation of MAIAC AOD

We used the full residual deep network to impute massive missing values of daily MAIAC AOD for 2015 in the Beijing-Tianjin-Tangshan area.

Our method on average achieved R^2 of 0.95 ranging from 0.71 to 0.99. Fig. 9 shows the grid surfaces of AOD before (a and c with massive missing values) and after (b and d with imputed values) imputation on two typical days: a warm day (05/13/2015) and a cool day (10/06/2015). The two days were selected because they had representative missingness (>50%) and reflect the AOD distributions in warm and cold seasons, respectively. Our results show that the full residual deep network reliably imputed missing AOD values with smooth variation in space and reasonable spatial patterns.

For the time series of imputed AOD data, we found excellent agreement between the imputed values and the ground truth AOD at two AERONET sites (Supplementary Fig. S4 for their locations and Fig. S9 for their scatter plots; Fig. 10 for the time series of the residuals between observed and predicted values): Pearson correlation of 0.93 with statistical significance ($p-value$<2.2e-16), R^2 of 0.81–0.84, and RMSE of 0.20–0.21. The means of the residuals were close to 0 with no significant pattern in their time series plots, indicating that our method well captured temporal variability of AOD estimates.

D. Spatiotemporal Prediction of $PM_{2.5}$

Our approach achieved the state-of-the-art accuracy for spatiotemoral estimation of $PM_{2.5}$ (for independent test, R^2: 0.88; RMSE: 24.01 μg/m^3) comparing to the results reported in the related literature [47]. Predicted daily $PM_{2.5}$ surfaces (spatial resolution: 1 km) on two typical days, 05/13/2015 and 10/06/2015 (Fig. 11) showed different spatial and temporal patterns, with higher concentrations in the cool season than that in the warm season and higher concentrations in the eastern region in the warm season and in the inner middle region in the cool season.

Additional independent test showed excellent agreement between measured and modeled daily $PM_{2.5}$ based on the monitoring data from the US embassy in Beijing (Fig. 12 and Supplementary Fig. S10), with Pearson's correlation of 0.99 ($p-value$<2.2e-16). The time series residual plot showed higher residuals in winter than those in the other seasons. Although spatial and temporal autocorrelations were not directly embedded in our model, Moran's I [67] of the daily residuals of the measurements showed complete spatial randomness for 150 days ($p-value$>0.05) and partial randomness (small mean Moran's I: 0.09) for the rest of days, illustrating that the spatial autocorrelations were well captured. The use of spatially and/or temporally varying explanatory variables (coordinates, elevation, meteorology, and measured and imputed AOD) likely helped to capture the spatiotemporal autocorrelations.

E. Method Comparisons

Besides the comparison of our method with non-residual plain network to show the contribution of full residual connections, we compared our method with three other methods, namely CRA, GAM and XGBoost. CRA is a neural network using cascaded residual shallow autoencoders [18]. GAM is a classical non-linear regression method, commonly used in AOD and $PM_{2.5}$ estimation. XGBoost is a typical advanced machine learner with solid performance in many practical applications [68] including $PM_{2.5}$ prediction [54]. Since XGBoost is an ensemble learner, we compared it to bootstrap aggregating (bagging) of the full residual deep network for a more fair evaluation. The proposed full residual deep network performed better than CRA (20 shallow autoencoders cascaded in this method) and GAM (Table V and Supplementary Fig. S11 for AOD imputation). The bagging version of the full residual deep network (100 base models used) performed similarly in R^2 and RMSE as XGBoost. Although similar spatial distributions of $PM_{2.5}$ were predicted by our full residual deep network (Fig. 11) and XGBoost (Fig. 13), the surfaces predicted by our method showed more smooth and realistic spatial patterns, while the surfaces predicted by XGBoost had abrupt and unrealistic changes at certain locations. The comparisons suggest the state-of-the-art performance of our proposed method.

V. Discussion

A robust machine learning model for predicting spatiotemporal variables needs to capture complex non-linear relationships among spatially and/or temporally varying influential factors. For many real-world problems where CNN or deep plain MLP may not perform well (e.g. due to a limited sample size and/or weak translation invariance), with efficient nested residual connections across the encoder and the decoder, and latent representation extraction, our novel encoder-decoder full residual deep network can be well applied to model complex non-linear relationships even for data with a small sample size and/or with weak translation invariance. Through extensive model evaluation and comparisons, our method has been demonstrated to be efficient and robust in spatiotemporal modeling of continuous variables and can be applied to both small and large samples.

Although different from ResNet that commonly uses residual units, an encoder-decoder network with full residual connections has advantages over a network with no or limited residual connections because full residual connections across the encoder and the decoder can facilitate efficient information backpropagation between each shallow layer and its deep counterpart, as shown theoretically in this paper and supported by our test results on the number and placements of residual connections. We found that although activation functions such as ReLU and ELU usually work well for hidden layers, a linear activation function may work better for the output layer due to its high computing efficiency.

Our approach consistently achieved the state-of-the-art performance, with better performance than non-residual plain network in most cases (>80%). For AOD imputation and spatiotemporal $PM_{2.5}$ estimation, the full residual deep network performed considerably better than non-residual plain network, CRA and GAM; its test R^2 values were also better than those reported in many previous studies (for AOD: 0.80–0.86 vs. 0.18–0.44; for $PM_{2.5}$: 0.88 vs. 0.58–0.87). Particularly, our method had less bias in predicting extreme values

(both high and low ends) of the continuous target variables, which cannot be well predicted in typical regression models such as linear regression, GAM [69] or plain MLP. Although plain MLP was used in previous studies [43], [55] to achieve a competent performance, its performance was less desirable than our full residual deep network for spatiotemporal modeling of complex non-linear systems (e.g. AOD imputation and $PM_{2.5}$ predictions), likely due to accuracy degradation in deep layers. CRA used cascaded shallow autoencoders; the lack of deep layers in its autoeoconder may limit its ability to model complex non-linear relationships. GAM usually needs the data to satisfy a presumed probability distribution, which limits its generalization. In addition, GAM relies solely on polynomial functions for parameters, which may limit sharing and interactions of the parameters and hence its prediction power.

The bagging version of our full residual deep network achieved similar test R^2 and RMSE but generated smoother and more realistic spatial surfaces of $PM_{2.5}$ than the ensemble-based XGBoost decision tree model. XGBoost discretizes continuous variables in its decision trees, which may result in information loss and abrupt or unrealistic changes in continuous variables such as $PM_{2.5}$, especially in small samples. In contrast, our full residual deep network does not discretize data and keeps all the input information, thus the bagging version of this model can be more powerful than XGBoost for unbiased predictions of continuous variables.

VI. Conclusion

In this paper, we propose a full residual deep network nested in the encoder-decoder symmetrical architecture. The full residual connections can compensate error backpropagation through the long path of deep layers, reduce degradation of accuracy, and improve learning efficiency. Different from the cascaded residual autoencoder, our approach directly and fully embeds residual connections from the shallow layers to the deep layers of the network in an end-to-end way. It can efficiently learn and achieve an optimal solution despite increased network complexity, as demonstrated in our testing results. The proposed full residual deep network can be an efficient and powerful tool in a variety of applications that involve complex non-linear relationships, varying sample sizes (particularly small samples), and spatiotemporal modeling of variables with weak or little invariance to translation due to non-linear processes and multiple influential factors.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgment

The support of NVIDIA Corporation with the donation of the Titan Xp GPUs used for this research is gratefully acknowledged. The authors would like to thank the editors and reviewers for their constructive comments that have helped us improve this paper.

This work was supported in part by the Strategic Priority Research Program of Chinese Academy of Sciences Grant XDA19040501, in part by the National Natural Science Foundation of China under Grant 41471376, and in part by the National Institute of Environmental Health Sciences under Grant ES030353.

Biographies

Lianfa Li (M'20) received the the Ph.D. degree in geographical information science from the Chinese Academy of Sciences (CAS), Beijing, China, in 2006.

He is currently an associate Professor with the State Key Laboratory of Resources and Environmental Information Systems, Institute of Geographic Sciences and Nature Resources Research, CAS, Beijing. He has been a visiting senior researcher at the University of Southern California, engaged in geospatial big data mining and machine learning in environmental and health sciences. His current research interests include spatiotemporal deep learning, remote sensing information extraction, spatial statistics, and environmental statistics and modeling.

Ying Fang received the B.S. degree in geographical information system from Anhui University, He'fei, China, in 2016, and the M.S. degree in geographical information system from the Chinese Academy of Sciences (CAS), Beijing, China, in 2019, respectively.

She was engaged in geospatial data collection, implementation of spatial statistical modeling, validation and data analysis.

Jun Wu received the degree of Environmental Health Sciences from University of California, Los Angeles in 2004.

She has been a faculty at University of California, Irvine since 2006 and is currently a professor of Environmental Health with the Department of Environmental and Occupational Health. Her general research interests include population-based research of environmental exposure assessment, environmental epidemiology, and environmental health disparity. She is particularly interested in applying machine learning methods and big data in informing human environmental exposure assessment, time-activity tracking, and environmental epidemiological research.

Jinfeng Wang received the B.Sc. degree in geography from Shaanxi Normal University, Xi'an, China, in 1985, the M.Sc. degree in physics from the Institute of Glaciology and Geocryology, Chinese Academy of Sciences (CAS), Lanzhou, China, in 1988, and the Ph.D. degree in GISci from the Institute of Geography, CAS, Beijing, China, in 1991.

He is currently a Professor with the State Key Laboratory of Resources and Environmental Information Systems, Institute of Geographic Sciences and Nature Resources Research, CAS, Beijing. His research interests include spatial statistics and its application in geoscience and population health.

Yong Ge (M'14) received the Ph.D. degree in cartography and geographical information system from the Chinese Academy of Sciences (CAS), Bei'jing, China, in 2001.

She is currently a Professor with the State Key Laboratory of Resources and Environmental Information System, Institute of Geographical Sciences and Natural Resources Research, CAS. Her research interests broadly focus on the statistical aspects of spatial and spatio-temporal data. Her research topics range from spatial-temporal sampling, scaling, remote sensing image analysis and big data analysis. Currently she is the associate editor of the Computers and Geosciences and the editor board member of Spatial Statistics. She is also the guest editor of various special issues of journals. Dr. Ge has authored more than 180 papers in the field of spatial statistics, information extraction from remotely sensed imagery and related fields. She holds more than ten granted patents in improving the accuracy of information extraction from spatial data.

References

[1]. LeCun Y, Bengio Y, and Hinton G, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 5 2015. [PubMed: 26017442]

[2]. Rumelhart DE, Hinton GE, and Williams RJ, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533–536, 10. 1986.

[3]. Raina R, Madhavan A, and Ng AY, "Large-scale deep unsupervised learning using graphics processors," in Proc. 26th Ann. Int. Conf. Mach. Learn. ACM, 2009, pp. 873–880.

[4]. Glorot X, Bordes A, and Y. B, "Deep sparse rectifier neural networks," in Proc. 14th Int. Conf. on Artif. Intell. Stat., 4. 2011, pp. 315–323.

[5]. LeCun Y, Boser BE, Denker JS, Henderson D, Howard RE, Hubbard WE, and Jackel LD, "Handwritten digit recognition with a back-propagation network," in Proc. Adv. Neu. Inf. Proc. Sys, 1990, pp. 396–404.

[6]. Hochreiter S. and Schmidhuber J, “Long short-term memory,” Neu. Comput, vol. 9, no. 8, pp. 1735–1780, 1997.

[7]. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, and Bengio Y, “Generative adversarial nets,” in Proc. Adv. Neu. Inf. Proc. Sys, 2014, pp. 2672–2680.

[8]. Hinton G, “Deep belief networks,” Scholarpedia, vol. 4, no. 5, p. 5947, 2009.

[9]. Simonyan K. and Zisserman A. (2014) ”Very deep convolutional networks for large-scale image recognition”. [Online]. Available: https://arxiv.org/abs/1409.1556

[10]. Goodfellow I, Bengio Y, and Courville A, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[11]. Srivastava RK, Greff K, and Schmidhuber J. (2015) ”Highway networks”. [Online]. Available: https://arxiv.org/abs/1505.00387

[12]. He KM, Zhang XY, Ren SQ, and Sun J, “Identity mappings in deep residual networks,” Computer Vision - ECCV 2016, Pt Iv, vol. 9908, pp. 630–645, 2016.

[13]. Zhang Z, Liu Q, and Wang Y, “Road extraction by deep residual u-net,” IEEE Geosci. Remote. Sens. Lett, vol. 15, no. 5, pp. 749–753, 3. 2018.

[14]. Zhang Y, Li K, Li K, Wang L, Zhong B, and Fu Y, “Image super-resolution using very deep residual channel attention networks,” in Proc. Eur. Conf. Comp. Vis. (ECCV), 9. 2018, pp. 286–301.

[15]. Zhang Y, Tian Y, Kong Y, Zhong B, and Fu Y, “Residual dense network for image super-resolution,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 6. 2018, pp. 2472–2481.

[16]. Alexandre D, Chang C-P, Peng W-H, and Hang H-M, “An autoencoder-based learned image compressor: Description of challenge proposal by nctu,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops, 6. 2018, pp. 2539–2542.

[17]. Zhang J, Zheng Y, and Qi D, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in Proc. 31st AAAI Conf. Artif. Intell., 2. 2017, pp. 1655–1661.

[18]. Tran L, Liu X, Zhou J, and Jin R, “Missing modalities imputation via cascaded residual autoencoder,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 7. 2017, pp. 1405–1414.

[19]. Zini S, Bianco S, and Schettini R. (2019) ”Deep residual autoencoder for quality independent JPEG restoration”. [Online]. Available: https://arxiv.org/abs/1903.06117

[20]. Bengio Y, Courville Y, and Vincent P, “Representation learning: a review and new perspectives,” IEEE Trans. PAMI, special issue Learning Deep Architectures, vol. 35, no. 8, pp. 1798–1828, 3. 2013.

[21]. Allen B. (2017) ”Atmospheric aerosols: what are they, and why are they so important?”. Accessed: Mar. 10, 2019. [Online]. Available: https://www.nasa.gov/centers/langley/news/factsheets/Aerosols.html

[22]. EPA. (2015) ”Particulate matter emissions”. [Online]. Available: http://www.epa.gov/roe/

[23]. Saunders A, Oldenburg IA, Berezovskii VK, Johnson CA, Kingery ND, Elliott HL, Xie T, Gerfen CR, and Sabatini BL, “A direct GABAergic output from the basal ganglia to frontal cortex,” Nature, vol. 521, no. 7550, pp. 85–9, 5 2015. [PubMed: 25739505]

[24]. He KM, Zhang XY, Ren SQ, and Sun J, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 6. 2016, pp. 770–778.

[25]. Kingma P. and Welling M. (2013) ”Auto-encoding variational Bayes”. [Online]. Available: https://arxiv.org/abs/1312.6114

[26]. Liou CY, Cheng WC, Liou JW, and Liou DR, “Autoencoder for words,” Neurocomputing, vol. 139, pp. 84–96, 9. 2014.

[27]. Ronneberger O, Fischer P, and Brox T. (2015) ”U-Net: convolutional networks for biomedical image segmentation”. [Online]. Available: https://arxiv.org/abs/1505.04597

[28]. Badrinarayanan V, Kendall A, and Cipolla R, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell, vol. 39, no. 12, pp. 2481–2495, 2017. [PubMed: 28060704]

[29]. Kostadinov S. (2019) ”Understanding encoder-decoder sequence to sequence model”. [Online]. Available: https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346

[30]. Jolliffe T, Principal Component Analysis (second edition). New York, USA: Springer-Verlag, 2002.

[31]. Gerven v., “Computational foundations of natural intelligence,” Front. Comput. Neurosci, vol. 11, pp. 1–24, 12. 2017. [PubMed: 28163679]

[32]. Jegou H, Perronnin F, Douze M, Sanchez J, Perez P, and Schmid C,´ “Aggregating local image descriptors into compact codes,” IEEE Trans. Pattern Anal. Mach. Intell, vol. 34, no. 9, pp. 1704–1716, 9. 2012. [PubMed: 22156101]

[33]. Veit A, Wilber M, and Belongie S, “Residual networks behave like ensembles of relatively shallow networks,” in NIPS’16 Proc. 30th Int. Conf. Neu. Inform. Proc. Sys., 2016, pp. 550–558.

[34]. Zhang L, Zhang L, and Du B, “Deep learning for remote sensing data: A technical tutorial on the state of the art,” IEEE Geosci. Remote Sens. Mag, vol. 4, no. 2, pp. 22–40, 6. 2016.

[35]. Lea C, Vidal R, Reiter A, and Hager GD, “Temporal convolutional networks: A unified approach to action segmentation,” in Proc. Eur. Conf. Comput. Vis. Springer, 10. 2016, pp. 47–54.

[36]. Taylor GW, Fergus R, LeCun Y, and Bregler C, “Convolutional learning of spatio-temporal features,” Computer Vision - Eccv 2010, Pt Vi, vol. 6316, pp. 140–153, 2010.

[37]. Zhang SH, Wu GH, Costeira JP, and Moura JMF, “FCN-rLSTM: Deep spatio-temporal neural networks for vehicle counting in city cameras,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 7. 2017, pp. 3687–3696.

[38]. Zhang R, Li N, Huang S, Xie P, and Jiang H, “Automatic prediction of traffic flow based on deep residual networks,” in Int. Conf. Mobi. Ad-Hoc Sensor Nets. Springer, 8. 2017, pp. 328–337.

[39]. Xi G, Yin L, Li Y, and Mei S, “A deep residual network integrating spatial-temporal properties to predict influenza trends at an intra-urban scale,” in Proc. 2nd ACM SIGSPATIAL Int. Wksp. on AI for Geo. Knowl. Discov. ACM, 11. 2018, pp. 19–28.

[40]. Du B, Peng H, Wang S, Bhuiyan MZA, Wang L, Gong Q, Liu L, and Li J, “Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction,” IEEE Trans. Intell. Transp. Syst, pp. 1–14, 3. 2019.

[41]. Kloog I, Nordio F, Coull BA, and Schwartz J, “Incorporating local land use regression and satellite aerosol optical depth in a hybrid model of spatiotemporal $PM_{2.5}$ exposures in the Mid-Atlantic states,” Environ. Sci. Technol, vol. 46, no. 21, pp. 11913–11921, 11. 2012. [PubMed: 23013112]

[42]. Lv B, Hu Y, Chang HH, Russell AG, and Bai Y, “Improving the accuracy of daily $PM_{2.5}$ distributions derived from the fusion of ground-level measurements with aerosol optical depth observations, a case study in North China,” Environ. Sci. Technol, vol. 50, no. 9, pp. 4752–4759, 4. 2016. [PubMed: 27043852]

[43]. Di Q, Kloog I, Koutrakis P, Lyapustin A, Wang Y, and Schwartz J, “Assessing $PM_{2.5}$ exposures with high spatiotemporal resolution across the continental united states,” Environ. Sci. Technol, vol. 50, no. 9, pp. 4712–4721, 5 2016. [PubMed: 27023334]

[44]. Li S, Chen L, Tao J, Han D, Wang Z, Su L, Fan M, and Yu C, “Retrieval of aerosol optical depth over bright targets in the urban areas of North China during winter,” Sci. China Earth Sci, vol. 55, no. 9, pp. 1545–1553, 5 2012.

[45]. Van Donkelaar A, Martin RV, Brauer M, Kahn R, Levy R, Verduzco C, and Villeneuve PJ, “Global estimates of ambient fine particulate matter concentrations from satellite-based aerosol optical depth: development and application,” Environ. Health Perspect, vol. 118, no. 6, pp. 847–855, 6. 2010. [PubMed: 20519161]

[46]. Xiao Q, Wang Y, Chang HH, Meng X, Geng G, Lyapustin A, and Liu Y, “Full-coverage high-resolution daily $PM_{2.5}$ estimation using MAIAC AOD in the Yangtze River Delta of China,” Remote Sens. Environ, vol. 199, pp. 437–446, 10. 2017.

[47]. Li L, Zhang J, Meng X, Fang Y, Ge Y, Wang J, Wang C, Wu J, and Kan H, “Estimation of $PM_{2.5}$ concentrations at a high spatiotemporal resolution using constrained mixed-effect bagging models with MAIAC aerosol optical depth,” Remote Sens. Environ, vol. 217, pp. 573–586, 9. 2018.

[48]. Bai Y, Wu L, Qin K, Zhang Y, Shen Y, and Zhou Y, “A geographically and temporally weighted regression model for ground-level $PM_{2.5}$ estimation from satellite-derived 500 m resolution AOD,” Remote Sens, vol. 8, no. 3, p. 262, 3. 2016.

[49]. Guo Y, Tang Q, Gong D-Y, and Zhang Z, “Estimating ground-level $PM_{2.5}$ concentrations in Beijing using a satellite-based geographically and temporally weighted regression model,” Remote Sens. Environ, vol. 198, pp. 140–149, 8. 2017.

[50]. Lee M, Kloog I, Chudnovsky A, Lyapustin A, Wang Y, Melly S, Coull B, Koutrakis P, and Schwartz J, “Spatiotemporal prediction of fine particulate matter using high-resolution satellite images in the Southeastern US 2003–2011,” J. Expo. Sci. Environ. Epidemiol, vol. 26, no. 4, pp. 377–384, 12. 2016. [PubMed: 26082149]

[51]. Xie Y, Wang Y, Zhang K, Dong W, Lv B, and Bai Y, “Daily estimation of ground-level $PM_{2.5}$ concentrations over Beijing using 3 km resolution MODIS AOD,” Environ. Sci. Technol, vol. 49, no. 20, pp. 12280–12288, 8. 2015. [PubMed: 26310776]

[52]. Hu X, Waller LA, Lyapustin A, Wang Y, Al-Hamdan MZ, Crosson WL, Estes MG Jr, Estes SM, Quattrochi DA, Puttaswamy SJ et al. , “Estimating ground-level $PM_{2.5}$ concentrations in the Southeastern United States using MAIAC AOD retrievals and a two-stage model,” Remote Sens. Environ, vol. 140, pp. 220–232, 1. 2014.

[53]. Hu X, Belle JH, Meng X, Wildani A, Waller LA, Strickland MJ, and Liu Y, “Estimating $PM_{2.5}$ concentrations in the conterminous United States using the random forest approach,” Environ. Sci. Technol, vol. 51, no. 12, pp. 6936–6944, 5 2017. [PubMed: 28534414]

[54]. Zamani Joharestani M, Cao C, Ni X, Bashir B, and Talebiesfandarani S, “$PM_{2.5}$ prediction based on random forest, XGBoost, and deep learning using multisource remote sensing data,” Atmos, vol. 10, no. 7, p. 373, 7. 2019.

[55]. Feng X, Li Q, Zhu Y, Hou J, Jin L, and Wang J, “Artificial neural networks forecasting of $PM_{2.5}$ pollution using air mass trajectory based geographic model and wavelet transformation,” Atmos. Environ, vol. 107, pp. 118–128, 4. 2015.

[56]. Wiki. (2015) ”Residual neural network”. [Online]. Available: https://en.wikipedia.org/wiki/Residual_neural_network

[57]. Adaloglou N. (2020) ”Intuitive explanation of skip connections in deep learning”. [Online]. Available: https://theaisummer.com/skip-connections/

[58]. Zou H. and Hastie T, “Regularization and variable selection via the elastic net,” J. R. Stat. Soc. B, vol. 67, no. 2, pp. 301–320, 2005.

[59]. Szegedy C, “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1–9.

[60]. Baydin AG, Pearlmutter B, Radul AA, and Siskind J, “Automatic differentiation in machine learning: a survey,” J. Mach. Learn. Res, vol. 18, pp. 1–43, 2018.

[61]. Clevert D-A, Unterthiner T, and Hochreiter S. (2015) ”Fast and accurate deep network learning by exponential linear units (elus)”. [Online]. Available: https://arxiv.org/abs/1511.07289

[62]. LeCun Y, Bottou L, Orr GB, and Mller K, Efficient Backpropopagation. New York, USA: Springer, 1998, pp. 9–50.

[63]. Ioffe S. and Szegedy C. (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. [Online]. Available: https://arxiv.org/abs/1502.03167

[64]. Langkvist M, Karlsson L, and Loutfi A, “A review of unsupervised¨ feature learning and deep learning for time-series modeling,” Pattern Recognit. Lett, vol. 42, pp. 11–24, 2014.

[65]. Kingma DP and Ba J. (2014) ”Adam: A method for stochastic optimization”. [Online]. Available: https://arxiv.org/abs/1412.6980

[66]. He K, Zhang X, Ren S, and Sun J, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1026–1034.

[67]. Li H, Calder CA, and Cressie N, “Beyond Moran’s I: Testing for spatial dependence based on the spatial autoregressive model,” Geogr. Anal, vol. 39, no. 4, pp. 357–375, 9. 2007.

[68]. Chen T. and Guestrin C, “Xgboost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Disc. Data Min., 8. 2016, pp. 785–794.

[69]. Bruce A. and Bruce P, Practical Statistics for Data Scientists. Sebastopol, CA, USA: O'Reilly Media, Inc., 2017.

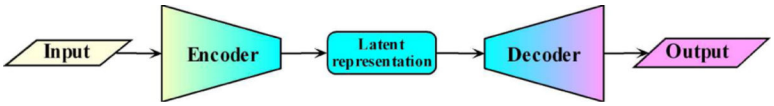


Fig. 1.
General encoder-decoder architecture.

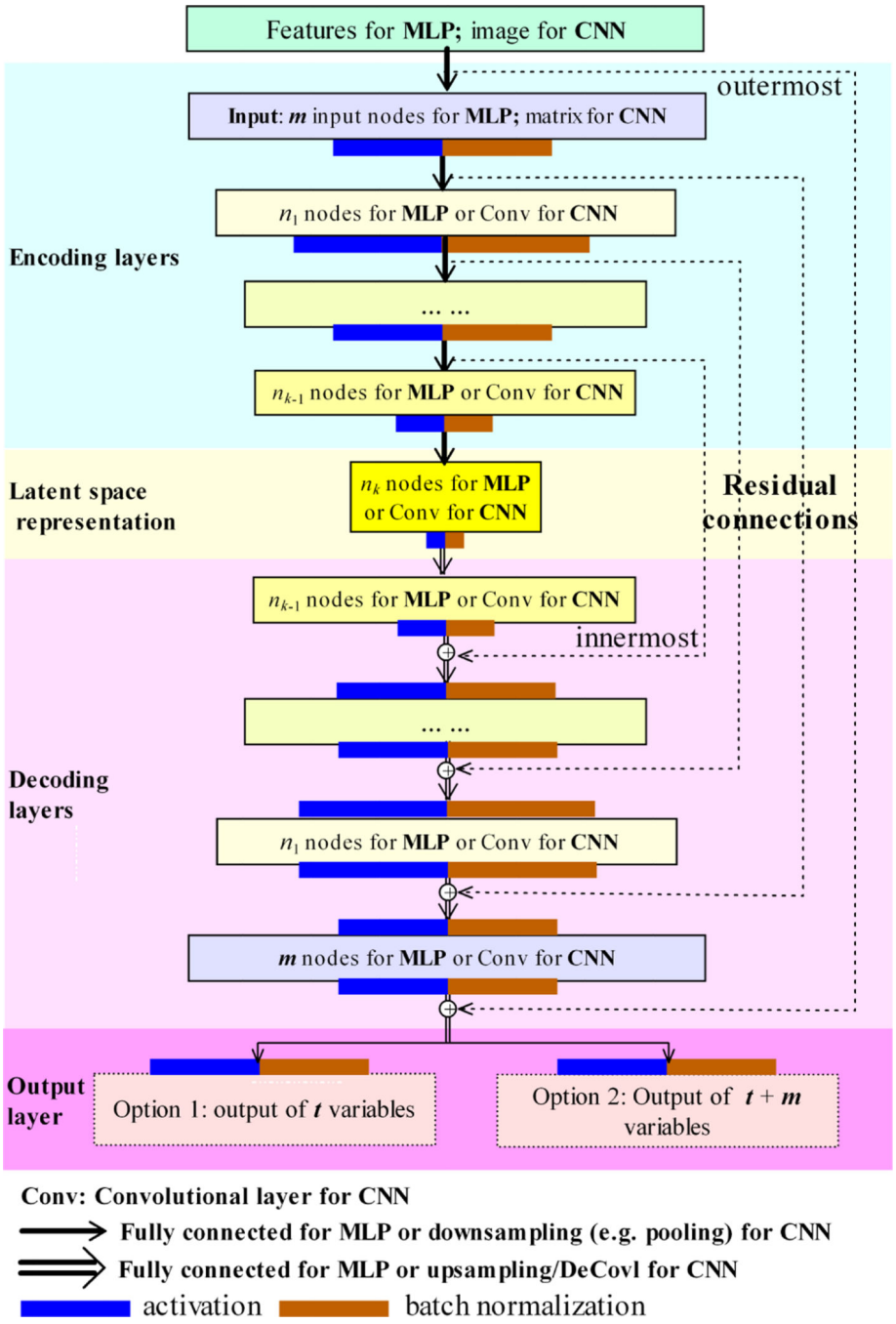


Fig. 2.
Architecture of a typical encoder-decoder deep residual network for MLP and CNN.

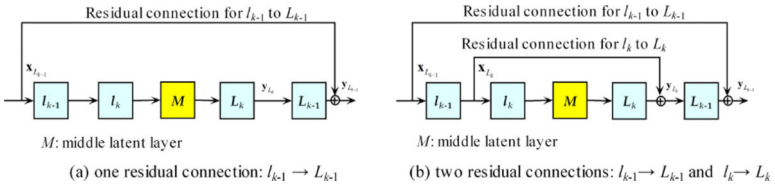


Fig. 3.

One residual connection (a) vs. two residual connections (b) surrounding the latent layer.

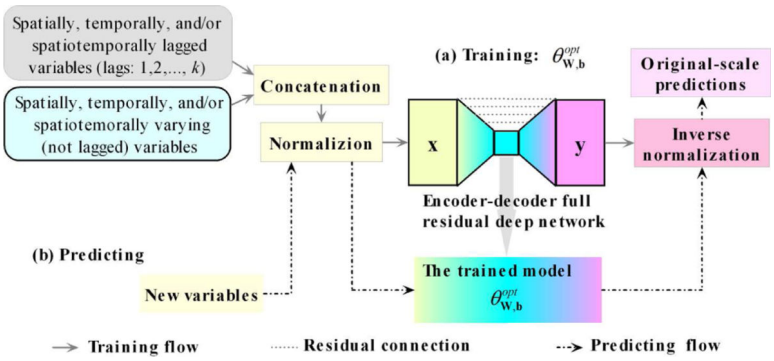
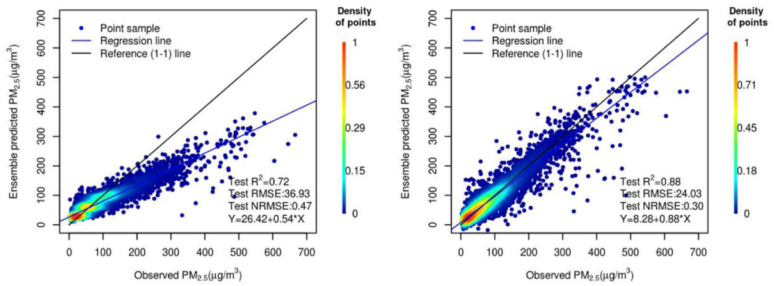


Fig. 4.
Training and predicting.



(a) Non-residual plain network (b) Full residual deep network

Fig. 5.

Scatter plots with sample density coloring for the predicted vs. observed $PM_{2.5}$

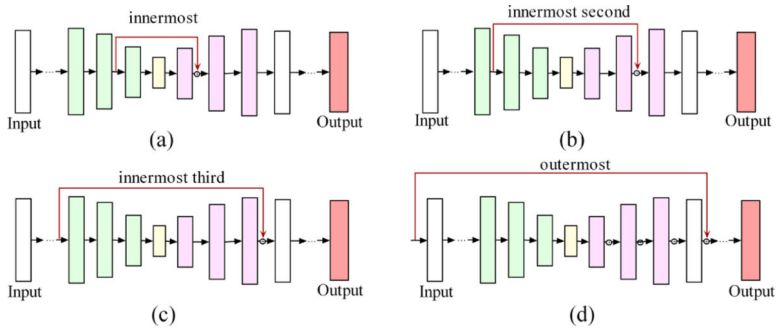


Fig. 6.
Networks with one residual connection at different placements.

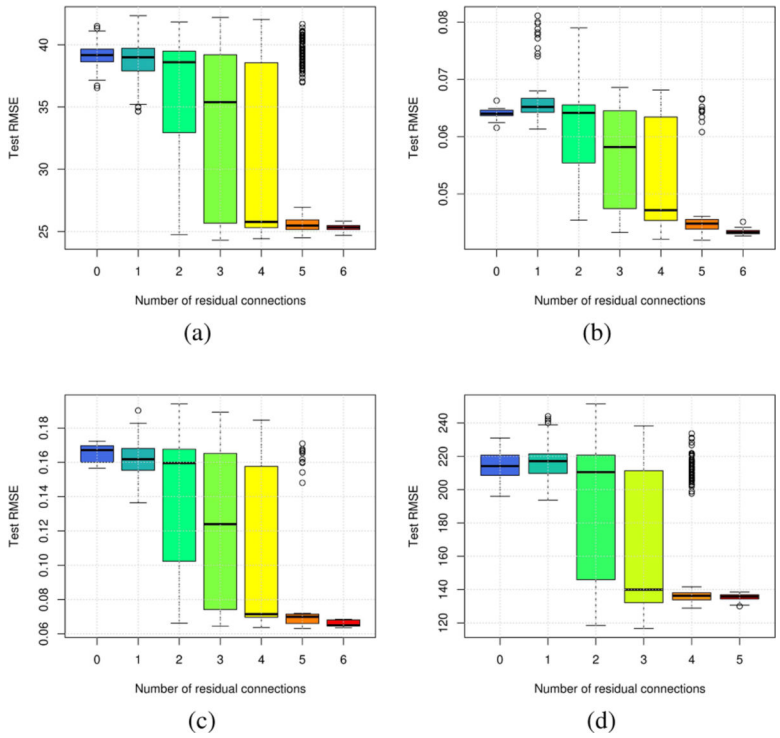


Fig. 7.

Statistical boxplots of test RMSE for the networks with different numbers of residual connections ((a) the simulated data; (b) MAIAC AOD of 05/13/2015; (c) MAIAC AOD of 10/06/2015; (d) $PM_{2.5}$).

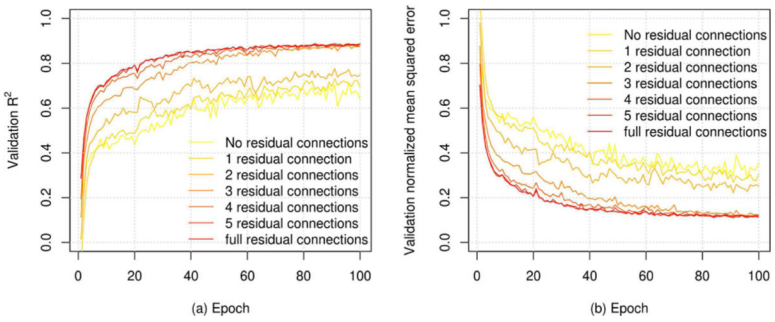


Fig. 8.

Learning curves (a: validation R^2; b: validation normalized mean squared error) for different numbers of residual connections for estimation of $PM_{2.5}$.

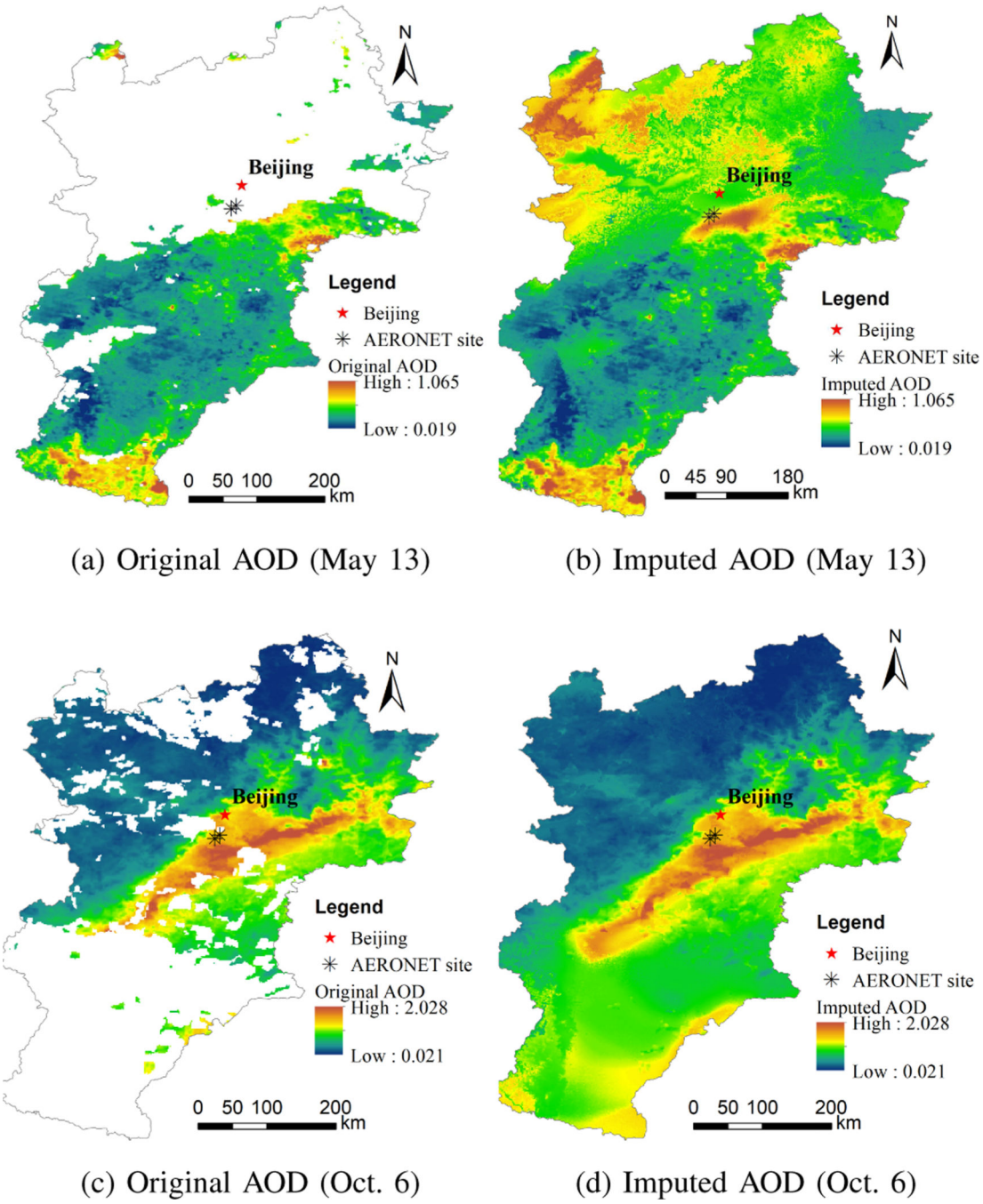


Fig. 9.

The MAIAC AOD surfaces of the original incomplete data (a and c) and the complete data after imputation (b and d) for two typical days of 2015 in the study region.

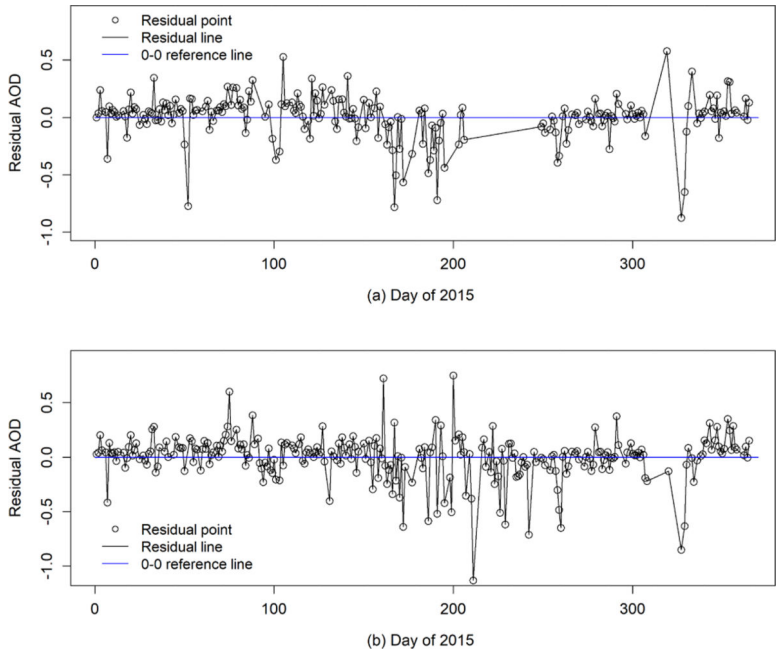


Fig. 10.

Time series of the residuals of AOD at two AERONET monitoring stations in Chaoyang (a) and Haidian (b) of Beijing.

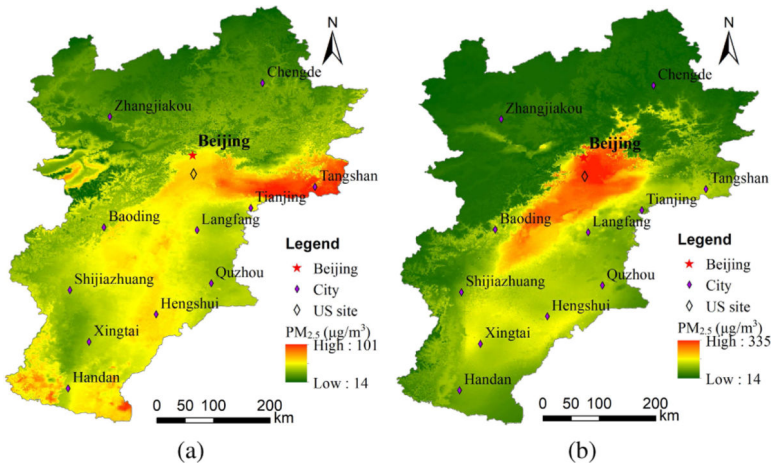


Fig. 11.

The $PM_{2.5}$ surfaces in Beijing-Tianjin-Tangshan area predicted by the full residual deep network for two days of warm season (05/13/2015, a) and cool season (10/06/2015, b).

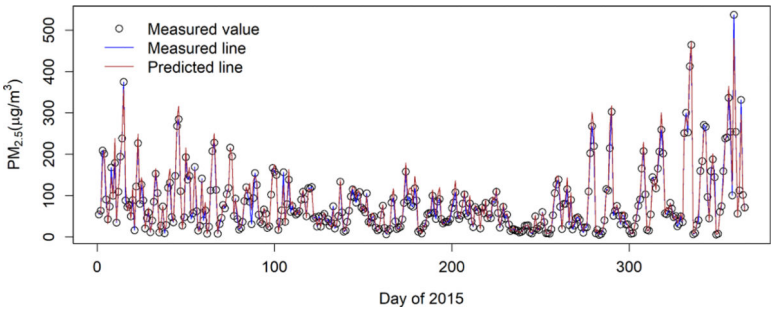


Fig. 12.

The 2015-day time series of the observed and predicted $PM_{2.5}$ for the US embassy monitoring station.

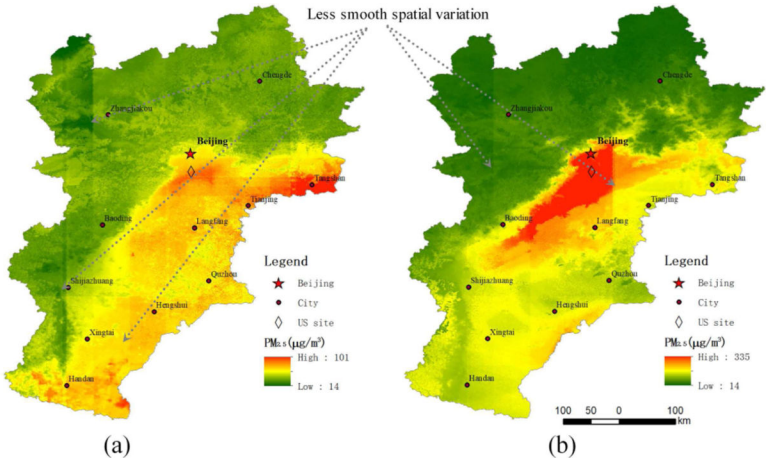


Fig. 13.

The $PM_{2.5}$ surfaces in Beijing-Tianjin-Tangshan area predicted by XGBoost for two typical days of warm season (05/13/2015, a) and cool season (10/06/2015, b).

TABLE I:

DATASETS FOR GENERAL TEST, IMPUTATION AND SPATIOTEMPORAL ESTIMATION

Study case		Domain	Target	Output type	#Fea[a]	#Samp[b]
Simulated dataset		Simulated	Regression	Continuous numerical values	8	1000
UCI Dataset[c]	ADULT	Households	Classification	Binary	123	5000
	Heart disease (Cleveland)	Life	Classification	Binary	14	303
	Abalone	Life	Classification	Category (1–8, 9–10, 11)[d]	8	4177
	Combined cycle power plant	Energy	Regression	Continuous numerical values	4	1030
	Wine quality	Business	Regression	Continuous numerical values	12	6497
	Airfoil self-noise	Physical	Regression	Continuous numerical values	6	1503
MAIAC AOD	365 sets of samples (each day for 2015)	Remote sensing	Imputation	Continuous numerical values	17	34842–732294
$PM_{2.5}$	One dataset for 2015	Environment	Spatiotemporal estimation	Continuous numerical values	24	33118

[a]Number of features (predictors);

[b]Number of samples;

[c]Datasets from the UCI benchmark repository of machine learning (http://archive.ics.uci.edu/ml);

[d]The target variable is classified to 3 categories according to the intervals.

TABLE II:

TEST PERFORMANCES FOR THE MODELS WITH A SINGLE RESIDUAL CONNECTION AT DIFFERENT PLACEMENTS

Res[a]	Simulated Data		MAIAC AOD 05/13/2015		10/06/2015		$PM_{2.5}$	
	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE
Innermost	0.51	233	0.68	0.06	0.88	0.14	0.68	39
Second innermost	0.59	214	0.70	0.06	0.86	0.14	0.68	39
Third innermost	0.60	212	0.69	0.06	0.87	0.14	0.68	39
Fourth innermost	0.59	213	0.69	0.06	0.87	0.14	0.68	39
Fifth innermost	-[b]	-	0.69	0.06	0.88	0.14	0.68	39
Outermost	0.59	213	0.43	0.08	0.90	0.12	0.73	36

[a] A single residual connection;

[b] - indicates unavailability of the result.

TABLE III:

MEAN TEST PERFORMANCES FOR THE MODELS WITH DIFFERENT NUMBERS OF RESIDUAL CONNECTIONS

NR[a]	#Sce[b]	#Model[c]	Simulated Data		MAIAC AOD 05/13/2015		10/06/2015		$PM_{2.5}$	
			R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE
0	1	100	0.59	213	0.70	0.06	0.86	0.15	0.68	39
1	6	600	0.58	217	0.64	0.07	0.88	0.14	0.69	38
2	15	1500	0.66	191	0.70	0.06	0.90	0.12	0.72	36
3	20	2000	0.74	166	0.75	0.05	0.92	0.11	0.76	33
4	15	1500	0.79	150	0.79	0.05	0.94	0.09	0.80	30
5	6	600	0.84	135	0.82	0.04	0.96	0.08	0.84	27
6	1	100	-[d]	-	0.85	0.04	0.98	0.06	0.88	24

[a] Number of residual connections;

[b] Number of scenarios (different placements for a fixed number of residual connections);

[c] Total number of models (100 models trained for each scenario);

[d] - indicates unavailability of the result. Here just show the number of scenarios and total number of models for AOD and $PM_{2.5}$.

TABLE IV:

TEST PERFORMANCES FOR THE MODELS WITH DIFFERENT SCALES OF NETWORK

Scale[a]	Simulated Data		MAIAC AOD 05/13/2015		10/06/2015		$PM_{2.5}$	
	MLP[b]	FRDN[c]	MLP	FRDN	MLP	FRDN	MLP	FRDN
Small	0.59	0.84	0.84	0.85	0.97	0.97	0.68	0.85
Moderate	0.81	0.85	0.85	0.87	0.97	0.97	0.81	0.88
Moderately large	0.86	0.87	0.86	0.89	0.97	0.98	0.86	0.87
Large	0.76	0.88	0.85	0.89	0.96	0.98	0.76	0.88

[a]Defined according to the number of hidden layers; increasing in sequence from small, moderate to large [small: 4 encoding layers (12 layers in total); moderate: 5 encoding layers (14 layers in total); moderately large: 10 encoding layers (24 layers in total); large: 15 encoding layers (34 layers in total)];

[b]Plain MLP;

[c]Full residual deep network (FRDN).

TABLE V:

TEST PERFORMANCES FOR MULTIPLE METHODS

Model	Simulated Data		MAIAC AOD 05/13/2015 R2		10/06/2015		$PM_{2.5}$	
	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE
Bagging of full RDN[a]	0.89	127	0.90	0.031	0.98	0.051	0.90	22.41 μg/m^3
full RDN	0.84	133	0.87	0.040	0.98	0.054	0.88	24.01 μg/m^3
CRA	0.68	189	0.80	0.054	0.95	0.084	0.74	32.11 μg/m^3
GAM	0.62	199	0.61	0.069	0.89	0.131	0.49	50.85 μg/m^3
XGBoost	0.89	128	0.89	0.034	0.98	0.050	0.90	22.40 μg/m^3

[a]Bootstrap aggregating (Bagging) of full residual deep network (RDN).