
Monidipa Das, Student Member, IEEE, and Soumya K. Ghosh, Member, IEEE

Abstract—With the advent of advanced remote sensing technologies in past few decades, acquiring higher resolution satellite images has become easier and cheaper in recent days. However, on the other hand, it has offered a big challenge to the remote sensing community in smart image interpretation from such huge volume of data. Deep learning, which offers efficient algorithms for extracting multiple levels of feature abstractions, may be suitable to serve the purpose. This letter presents a deep learning approach (Deep-STEP) for spatiotemporal prediction of satellite remote sensing data. The proposed learning architecture is derived from a deep stacking network, consisting of a stack of multilayer perceptron, each of which models the spatial feature of the associated region at a particular time instant. The proposed method has been demonstrated on normalized difference vegetation index (NDVI) data sets, derived from satellite remote sensing imagery, containing several thousands to millions of pixels/records. The experimental results (related to NDVI prediction) reveal that the proposed architecture exhibits fairly satisfactory performance with promising learning capabilities.

Index Terms—Deep learning, deep stacking network (DSN), satellite remote sensing imagery, spatiotemporal prediction.

I. INTRODUCTION

THE recent explosive growth of spatiotemporal data in various domains has led to an immense interest in advanced spatiotemporal data mining techniques to automate the discovery of spatiotemporal knowledge from such huge data sets.

However, one of the major challenges here is the huge volume of the data that makes it difficult to extract the useful and interesting features. Prediction of satellite imagery is a prominent example in this regard. For instance, analysis of a typical Landsat Thematic Mapper (TM) scene (170 km × 183 km image containing seven spectral bands with a spatial resolution of 30 m) needs to deal with more than $2 \times 10^8$ pixels.1

Though the techniques, such as spatiotemporal autoregressive regression, spatiotemporal kriging, hierarchical dynamic spatiotemporal models, and graph-based models (MRF/CRF), are most commonly used for spatiotemporal prediction purpose [1], mining useful information from remote sensing data sets that contain millions of pixels is a major issue in these techniques [2]. Besides, the two major limitations of the graph-based models are: 1) the probability computation and parameter estimation in such models are very difficult and 2) these models and even the deep belief networks are not designed for parallel training capability and scalability. Further, the techniques based on fractal or multifractal analysis [3], [4] are also not suitable in this scenario, since the temporal frequency of such data sets is not always available at daily scale.

In contrast to the above-mentioned methods, our present work proposes a supervised deep learning approach (Deep-STEP) as a model for spatiotemporal prediction with satellite remote sensing data. The availability of the huge data set provides a suitable environment to Deep-STEP for hierarchically modeling high-level abstractions in the data. Moreover, with the use of Lagrangian multiplier method during weight learning, the approach can be easily extended to parallel version, and thereby becomes scalable for working with very large data sets.

A. Problem Statement and Contributions

Given a set of spatiotemporal raster data (e.g., sequence of remote sensing images) over a variable $v$ for the time instants $t_1, t_2, \ldots, t_n$, the problem is to predict $v$ for the time instant $t_{(n+1)}$ in the same spatiotemporal framework.

For that purpose, this letter proposes a novel approach, namely, Deep-STEP, which is derived from deep stacking network (DSN) model [5]. The deep learning techniques have been applied in natural language processing [6], audio and speech processing [7], and even in feature selection and classification in the field of remote sensing [8]–[10]. However, to the best of our knowledge, it has not been explored much for the spatiotemporal prediction of remote sensing data. The proposed Deep-STEP approach formulates the computation in DSN in a form of spatiotemporal feature modeling. In order to achieve this, unlike each time copying the bottommost feature set in standard DSN architecture, in Deep-STEP, we introduce a new temporally evolved feature set at the input layer of each module. It helps to incorporate the temporal change along with the spatial feature learning at each hidden layer. The Deep-STEP approach has been empirically compared with the standard DSN, multilayer perceptron (MLP), and nonlinear autoregressive neural network (NARNET) learning techniques, for normalized difference vegetation index (NDVI) prediction over four zones in Kharagpur (India). The prediction results

Manuscript received April 11, 2016; revised August 26, 2016; accepted September 19, 2016. Date of publication November 22, 2016; date of current version December 7, 2016.

The authors are with the Department of Computer Science and Engineering, IIT Kharagpur, Kharagpur 721302, India (e-mail: monidipadas@hotmail.com; skg@iitkgp.ac.in)

Digital Object Identifier 10.1109/LGRS.2016.2619984

1http://landsat.usgs.gov/band_designations_landsat_satellites.php

II. DEEP-STEP: PROPOSED DEEP LEARNING APPROACH FOR SPATIOTEMPORAL PREDICTION

The deep network model in the proposed Deep-STEP approach is derived from DSNs, which have been originally designed to make the deep learning process scalable. The details of DSN can be found in [5]. Similar to DSN, the central idea in the Deep-STEP architecture is the concept of stacking, where simple modules of functions are composed first and then these are stacked on top of each other for learning more complex functions.

However, the two major differences of the proposed Deep-STEP with the typical DSN are: 1) instead of each time copying the bottommost feature set, in Deep-STEP, we introduce a newly evolved feature set at the input layer of each module and 2) the output of each module is only stacked with a new input feature set of the same dimension in order to form the input layer of the immediate top module. That is, unlike the original DSN, here the input–output stack size is restricted to two sets of features. The first characteristic assists Deep-STEP in incorporating the temporal change along with the spatial feature learning at each hidden layer, whereas the second characteristic helps in reducing the complexity of the network architecture as well as the learning time.

The simplified block diagram of the proposed Deep-STEP approach is shown in Fig. 1. As depicted in the figure, the overall process passes through three key steps: 1) feature set preparation; 2) spatiotemporal feature learning; and 3) prediction. Each of these steps has been illustrated below.

A. Feature Set Preparation

In Deep-STEP, the data set (for training/testing) is prepared in such a way that each of the data record represents a pixel in terms of its spatiotemporal features. The feature set preparation is based on the assumption that the intensity of each pixel \( P(x, y, t) \) in the image raster can be modeled as a function of intensity of the neighboring pixels in space and time [12]. Mathematically, it can be expressed as follows:

\[
P(x, y, t) = \psi(P(x + \Delta x_i, y + \Delta y_j, t + \Delta t_k))
\]  

where \((\Delta x_i, \Delta y_j, \Delta t_k)\) signifies the neighborhood coverage in space and time with \(\Delta t_k < 0\).

Now, since our deep learning architecture takes into account the temporal evolution of each pixel, the feature set for a pixel \(P(x, y, t)\) at time \(t\) is prepared only with the neighboring pixels at time \(t - 1\) (including itself). Therefore, considering maximum spatial coverage of neighbor to be \(s\) in both the direction of \(x\) and \(y\), the feature set of \(P\) becomes

\[
\{P(x - s, y - s, t - 1), \ldots, P(x, y, t - 1), \ldots, P(x + s, y + s, t - 1)\}.
\]

In the same fashion, input data set dimension \([N \times M]\) is prepared for each time instant \(t\) separately, where, \(M = (2s+1)^2\) is the size of input feature vector corresponding to each of the \(N\) observed pixels at time \(t\).

B. Spatiotemporal Feature Learning

The spatiotemporal feature learning in Deep-STEP is achieved through modulewise weight updating in the proposed model.

1) Deep Network Architecture: The detailed architecture of the deep network model used in Deep-STEP approach is shown in Fig. 1. In the proposed deep architecture, each of the modules (denoted by distinct shade of patterns in Fig. 1) corresponds to a particular training year in sequence. Therefore, the total number of module is equal to the total number of training images. The concept of stacking has been illustrated in the figure by placing one module over the other, and arranging the output of the previous module and input of the current module in a pile. The bottommost module is associated with the training year which is at farthest temporal distance from the prediction year. The output (o/p) of each module (except the topmost one) is the learned spatial feature set for the corresponding time instant, whereas the input (i/p) to each module (except the bottommost one) is the combined raw spatial feature set as prepared in above-discussed manner (refer to Section II-A) and the spatial feature set learned by

![Fig. 1. Flow of the proposed Deep-STEP approach.](image-url)
the previous module. The input to the bottommost module is only the raw spatial feature set, and the output of the topmost module is the predicted value of the variable at a given pixel location.

Now, for any module in the deep network model, let the training vectors be denoted by \( X = [x_1, x_2, \ldots, x_N]^T \), in which each input vector \( x_i \) is of dimension \( D \) and is denoted by \( x_i = [x_{i1}, x_{i2}, \ldots, x_{id}] \), and \( N \) is the total number of training samples. Also let \( H = [h_1, h_2, \ldots, h_N]^T \) denote the activity matrix over all training samples in the hidden layer, \( L \) denotes the number of hidden units and \( C \) denotes the output vector dimension for any module. Then, as per the network architecture, the output of any module is \( y_i = h_i U^T \), where \( h_i = \sigma(x_i W^T) \) is the hidden-layer vector for sample \( i \); \( U \) is a \( [C \times L] \) dimensional weight matrix at the upper layer within the module; \( W \) is a \( [L \times D] \) dimensional weight matrix at the lower layer within the same module; and \( \sigma(\cdot) \) is a sigmoid function. The output of any module can be expressed in matrix form as follows:

\[
Y = \sigma(\sigma(XW^T)U^T). \quad (2)
\]

Here, \( W \) stores the weight between each pair of hidden unit and input unit, and similarly, \( U \) stores the weight between each pair of output unit and hidden unit.

Moreover, according to the proposed network architecture, the output of each module is stacked with a new input feature set of the same dimension, i.e., here the stack size is restricted to two. The value of \( D \) for the bottommost module is \( M \), whereas that for all the higher level modules is \( 2M \). In the architecture, the dimension of the hidden layer for any module is the same, i.e., \( L \), however, the output vector size \( C \) differs between the topmost module and the rest. The value of \( C \) for the topmost module is 1, but that for all the lower level modules is equal to \( L \). Similar to a DSN model, here the bias terms are implicitly represented in the above formulation if \( x_i \) and \( h_i \) are augmented with ones.

2) Deep Network Learning: The weight matrices \( W \) and \( U \) in DSN can be learned in various ways. In the proposed variant of deep network model, the \( W \) and \( U \) in each module are learned using conventional backpropagation method.

Given the target vectors over \( N \) samples, \( T = [t_1, t_2, \ldots, t_N]^T \), where each \( t_i = [t_{i1}, t_{i2}, \ldots, t_{iC}] \). Then, the cost function in feed-forward learning is estimated in the following manner:

\[
J = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} [t_{ij} \log y_{ij} + (1 - t_{ij}) \log (1 - y_{ij})] + R \quad (3)
\]

where \( R \) is the regularization term computed as follows:

\[
R = \frac{\lambda}{2N} \left[ \sum_{i=1}^{L} \sum_{j=1}^{D} u_{ij}^2 + \sum_{i=1}^{C} \sum_{j=1}^{L} w_{ij}^2 \right] \quad (4)
\]

where \( \lambda \) is the regularization parameter and \( u_{ij} \) and \( w_{ij} \) are the elements in \( i \)th row and \( j \)th column in the matrices \( U \) and \( W \), respectively.

Now, the gradient calculation for the cost function is determined through backpropagation in the following manner.

For the lower layer weights

\[
\frac{\partial J}{\partial W} = \frac{1}{N} \left[ \sum_{i=1}^{N} \left[ ((Y - T)U) \circ \sigma'(XW^T) \right]^T X \right] + \frac{\lambda}{N} W \quad (5)
\]

\[
= \frac{1}{N} \left[ \sum_{i=1}^{N} \left[ ((Y - T)U) \circ \sigma'(XW^T) \right]^T X + \lambda W \right] \quad (6)
\]

where

\[
\sigma'(XW^T) = \sigma(XW^T) \circ (1 - \sigma(XW^T)) \quad (7)
\]

and symbol \( \circ \) denotes elementwise multiplication.

Similarly, for the upper layer weights in the module

\[
\frac{\partial J}{\partial U} = \frac{1}{N} \left[ ((Y - T)^T \sigma(XW^T)) + \lambda U \right] \quad (8)
\]

\[
= \frac{1}{N} \left[ ((Y - T)^T \sigma(XW^T) + \lambda U) \right]. \quad (9)
\]

Once the output \( Y \) of a preceding (bottom) level module is obtained after learning, it is merged with the input feature set corresponding to the next time instant and then fed as the set of input vectors in the top level (succeeding) module (refer to Fig. 1). This helps to incorporate the temporal evolution of the pixels/data points in the feature learning process. Now, since the weight learning in Deep-STEP involves the sigmoid function \( \sigma(\cdot) \), the module output values are forced to be in the range \([0, 1]\). Therefore, in order to make the merged vectors consistent, each time the input data set \( X \) is normalized as follows:

\[
\text{normalized}_{x_{ij}} = \frac{(x_{ij} - \text{min}(X))}{(\text{max}(X) - \text{min}(X))} \quad (10)
\]

where \( \text{max}(X) \) and \( \text{min}(X) \) are the largest and smallest values in the matrix \( X \).

### C. Prediction

Now, based on the updated weight values and spatiotemporal features learned at the bottom levels, the topmost layer in the deep network generates all the \( N \) predicted values in a form of \([N \times 1]\) matrix, or in other words, generates an \( N \)-D vector \( Y \). Since the predicted values are obtained in normalized form within the range \([0, 1]\), these are further mapped to the original scale to get the actual prediction values.

### III. Experimental Evaluation

In this section, we validate the proposed deep learning approach (Deep-STEP) for spatiotemporal prediction of satellite remote sensing data.

#### A. Data Set and Study Area

In order to evaluate the method, a set of four time series of Landsat-7 TM-5 satellite imagery are collected from Land Process Distributed Active Archive Center of the United States Geological Survey (USGS) [11]. Each of these time series has a temporal frequency of one year and covers four different spatial zones in Kharagpur (central coordinate: [22.3302°N, 87.3236°E], India) during 2004–2011. The images have been selected after consultation with the domain experts, based on
the availability of the data in USGS site [11]. The selected images of 2004–07, 2009, and 2011 are from the month of March; that of the year 2008 is from the end of February; and the image of 2010 is from the beginning of April. According to the domain experts, as the months of February to May constitute the premonsoon season in Kharagpur (India), the vegetation cover remains more or less similar during this particular period of time. Moreover, before extracting the NDVI values, the raw satellite images for each study zone are passed through radiometric corrections. The Zone-1 has a 3 km × 3 km area, containing 10 000 pixels, whereas Zone-2 and Zone-4 have ≈10 km × 10 km area, each containing more than 100 000 pixels. Zone-3 is the largest region considered, which covers around 30 km × 30 km area, and contains 1 000 000 pixels. Initially, the data (raw data) was in the form of a set of seven spectral bands from which we extracted the NDVI2 information as the parameter to work with.

B. Experimental Setup

Experimentation has been carried out to predict the NDVI values in each of the four study zones for the year 2011, given the observed NDVI data of 2004 to 2010. The entire experiment has been performed using MATLAB 8.3.0.532 (R2014a) in Windows 2007 (64-b Operating System, 3.10 GHz CPU, 4.00 GB RAM), and ERDAS IMAGINE tool (version 9.2.1).3 ERDAS IMAGINE has been used to generate NDVI raster from the input raw satellite imagery, whereas the rest of the processing has been performed in MATLAB.

The performance of proposed Deep-STEP has been evaluated in comparison with three other models, namely, NARNET (from NN toolbox of MATLAB), MLP, and original DSN. The generic configuration for each of these models, as used in the experimentation, is given in Table I. The experiment has been carried out by considering spatial neighborhood coverage \( s = 1 \). Thus the number of input units for MLP is \((2 \times 1 + 1)^2 = 9\), and that for DSN and Deep-STEP in bottommost module is also nine.

Therefore, presently, the feature set for each target pixel is prepared with nine spatiotemporal neighborhood pixels (including itself) from the previous time instant, as per the concept of spatial neighborhood coverage (refer to Section II-A). Hence, according to the module architecture, the number of hidden units in each module is set to nine. On the other side, the input feature set for a top level (succeeding) module additionally contains nine more elements, representing some new spatiotemporal features learned from the lower level modules. While performing the cost optimization using gradient descent technique, the maximum number of iteration is considered to be 200, and the regularization parameter \( \lambda \) is set to 2. In the training phase, the images of 2004–2009 have been used to prepare feature set and the prediction has been made for the year 2010. On the other side, during testing, the feature set has been prepared by using the images of 2005–2010, and prediction has been made for the year 2011. In our experiment, linear stretching technique has been used to map the initially predicted NDVI values into the original scale. In the case of our present data set, the linear stretching is found to perform better than the nonlinear stretching techniques (namely, logarithmic stretching and power-law stretching). However, for other data sets, appropriate stretching techniques (linear/nonlinear) may be employed.

C. Results and Discussion

The results of comparative study with DSN, MLP, and NARNET, in terms of two error metrics [root mean square error (RMSE) and mean absolute error (MAE)], have been depicted in Fig. 2(a) and (b), respectively.
It is evident from the figures [refer to Fig. 2(a) and (b)] that the proposed Deep-STEP approach outperforms the others producing least RMSE and MAE in predicting the NDVI values for each of the study zones. The normalized error surfaces corresponding to the spatial distributions of NDVI for all the four zones in 2011 are depicted in Fig. 3. From the figure, it may be observed that compared with NARNET and MLP, the DSN produces significantly lesser error distribution. Further, the proposed Deep-STEP approach predicts even better in comparison with DSN, and thus, produces the least error distributions over the study zones. It not only proves the worth of considering deep network learning in prediction but also establishes the effectiveness of proposed Deep-STEP technique, which incorporates the deep learning from the perspective of a spatiotemporal feature learning model.

Comparative study has been carried out with respect to execution time as well. Table II summarizes the computing time of the methods under consideration. It is observed that the execution time for NARNET in predicting large-scale images is significantly higher, whereas that of MLP is notably lesser than the others. Further, it may be noted that the proposed Deep-STEP yields competitive and reasonable execution time, even slightly better than original DSN.

IV. CONCLUSION

This letter proposes a deep network learning approach (Deep-STEP) for spatiotemporal prediction of satellite remote sensing data. To the best of our knowledge, there is little work on application of deep learning in remote sensing image prediction. In our proposed Deep-STEP, the deep network model is derived from original DSN. However, the novelty lies here in formulating the DSN principle in terms of spatiotemporal feature learning and prediction. The proposed approach has been validated in predicting NDVI for 2011 over four spatial zones (in Kharagpur, India), each covering several thousands to millions of pixels. The comparative study with original DSN, MLP, and NARNET learning technique demonstrates the effectiveness and superiority of the proposed Deep-STEP approach, which always produces minimum prediction error for each of the study zones. In the future, the method can be extended to a parallel version in order to make it more scalable for very large remote sensing data sets.

TABLE II

<table>
<thead>
<tr>
<th>Image pixel count</th>
<th>Execution Time: [Training Time + Prediction Time] (in second)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NARNET</td>
</tr>
<tr>
<td>10000</td>
<td>43.1 x 10^4</td>
</tr>
<tr>
<td>100000</td>
<td>432 x 10^4</td>
</tr>
<tr>
<td>1000000</td>
<td>3996 x 10^4</td>
</tr>
</tbody>
</table>

It is evident from the figures [refer to Fig. 2(a) and (b)] that the proposed Deep-STEP approach outperforms the others producing least RMSE and MAE in predicting the NDVI values for each of the study zones. The normalized error surfaces corresponding to the spatial distributions of NDVI for all the four zones in 2011 are depicted in Fig. 3. From the figure, it may be observed that compared with NARNET and MLP, the DSN produces significantly lesser error distribution. Further, the proposed Deep-STEP approach predicts even better in comparison with DSN, and thus, produces the least error distributions over the study zones. It not only proves the worth of considering deep network learning in prediction but also establishes the effectiveness of proposed Deep-STEP technique, which incorporates the deep learning from the perspective of a spatiotemporal feature learning model.

Comparative study has been carried out with respect to execution time as well. Table II summarizes the computing time of the methods under consideration. It is observed that the execution time for NARNET in predicting large-scale images is significantly higher, whereas that of MLP is notably lesser than the others. Further, it may be noted that the proposed Deep-STEP yields competitive and reasonable execution time, even slightly better than original DSN.

IV. CONCLUSION

This letter proposes a deep network learning approach (Deep-STEP) for spatiotemporal prediction of satellite remote sensing data. To the best of our knowledge, there is little work on application of deep learning in remote sensing image prediction. In our proposed Deep-STEP, the deep network model is derived from original DSN. However, the novelty lies here in formulating the DSN principle in terms of spatiotemporal feature learning and prediction. The proposed approach has been validated in predicting NDVI for 2011 over four spatial zones (in Kharagpur, India), each covering several thousands to millions of pixels. The comparative study with original DSN, MLP, and NARNET learning technique demonstrates the effectiveness and superiority of the proposed Deep-STEP approach, which always produces minimum prediction error for each of the study zones. In the future, the method can be extended to a parallel version in order to make it more scalable for very large remote sensing data sets.

REFERENCES