LSTM Based Adaptive Filtering for Reduced Prediction Errors of Hyperspectral Images

Zhuocheng Jiang and W. David Pan
Dept. of Electrical and Computer Engineering
University of Alabama in Huntsville
Huntsville, AL 35899, USA
Email: zj0004@uah.edu; pand@uah.edu

Hongda Shen
*Bank of America Corporation**
New York, NY 10020, USA
Email: hongdadeeplearning@gmail.com

***Abstract*—While adaptive filtering has been widely used in predictive lossless compression of hyperspectral images, the prediction performance depends heavily on the filtering weights estimated in a step-by-step manner. Traditional filtering methods do not take into account the longer-term dependencies of the data to be predicted. Motivated by the effectiveness of recurrent neural networks in capturing data memory for time series prediction, we design LSTM (long short-term memory) networks that can learn the data dependencies indirectly from filter weight variations. We then use the trained networks to regulate the weights generated by conventional filtering schemes through a close-loop configuration. We compare the proposed method with two other memory-less algorithms, including the popular Least Mean Square (LMS) filtering method, as well as its variant based on the maximum correntropy criterion (MCC). Simulation results on two publicly available datasets show that the proposed LSTM based filtering method can achieve smaller prediction errors.**

***Index Terms*—Predictive compression, LSTM, Recurrent Neural Networks, Hyperspectral images, Least mean square**

I. Introduction

Efficient hyperspectral image compression is important for many remote sensing applications. Given that hyperspectral image sensor has only limited memory capacity, the ability to compress image cubes losslessly in real-time hyperspectral imaging system is extremely valuable for a wide range of applications, ranging from data transmission to data storage. To guarantee there is strictly no loss in the reconstructed data, in this work, we focus on lossless compression using predictive coding approaches, where we take advantage of the correlations of adjacent pixels [1] in the hyperspectral images. In these methods, pixel values are predicted from their causal contexts, with the prediction errors (also known as residuals) are coded by an entropy encoder. For hyperspectral data where contiguous spectral bands are highly correlated, linear prediction is usually used to model the correlated pixels in different spectral bands [2]. There have been several algorithms proposed for predictive lossless compression on hyperspectral image datasets [3], [4]. To improve the prediction accuracy, in our recent work, [5] proposed a two-stage prediction scheme by modeling the residuals with a mixture geometric distribution. [6] replaced the cost function of Least Mean Square (LMS) with Maximum Correntropy Criterion (MCC) to further reduce the entropy of residuals. It is worth noting that an LMS-based adaptive filtering method [7] has been chosen as the core predictor in the CCSDS standard for multispectral and hyperspectral data compression [8]. Besides linear predictive compression, nonlinear predictors such as 3-D CALIC [9], and context-based conditional average prediction (CCAP) [10] were also shown to provide good compression performance. Furthermore, transform-based methods intended for lossy compression [11] can be adapted for lossless compression.

While adaptive filtering has been widely used, its prediction performance depends heavily on the filtering weights estimated in a step-by-step manner [5], [7], [8]. These filtering methods do not take into account the longer-term dependencies of the data to be predicted. Motivated by the excellent performance of recurrent neural networks (RNNs) and their variants including LSTM (long short-term memory) [12] in time series prediction, in this work, we design LSTM networks that can learn the data dependencies indirectly from filter weight variations. The rest of this paper is organized as follows. Section 2 gives a brief introduction to the weight updating schemes in adaptive filtering and the basics of LSTM networks. Section 3 presents the proposed filtering method using LSTM networks. Experimental results are presented in Section 4. The paper is concluded in Section 5.

II. Adaptive Filtering and LSTM Networks

In adaptive filtering, the estimated pixel value was calculated as $W_n^T X_n$, where X_n and W_n are the context vector and the corresponding weight vector, respectively. The prediction error is $e_n = d_i - W_n^T X_n$, where d_i is the actual pixel value. Stochastic gradient descent is typically used with a small learning rate μ to maximize the cost function by iteratively adjusting the weight W as

$$W_{n+1} = W_n + \mu e_n X_n. \tag{1}$$

Our prior work [6] showed that MCC-LMS (Maximum Correntropy Criteria based Least Mean Square) could offer better compression on hyperspectral images than the conventional

*Note: This work is not associated with his current affiliation.

978-1-5386-4244-3/18/$31.00 ©2018 IEEE

LMS-based cost function. The correntropy [13] based weight updating at the n^{th} time instant can be written as:

$$W_{n+1} = W_n + \frac{\mu}{\sqrt{2\pi}\sigma^3}\exp\left(\frac{-{e_n}^2}{2\sigma^2}\right)e_n X_n, \tag{2}$$

where σ is the standard deviation of the normalized Gaussian kernel used.

It can be seen that filtering weights are updated in a step-by-step manner, without taking into account the memory of the dataset over longer discrete time steps. Our goal is to exploit the longer-time data dependencies to further improve the prediction accuracy. The idea is to learn the data dependencies using recurrent neural networks (RNNs), and use the trained networks to regulate the weights generated by conventional weight updating schemes. Recurrent neural networks differ from traditional feed-forward neural networks in that recurrent networks have an internal state that can represent context information. A RNN keeps information about past inputs for an amount of time that is not fixed *a priori*, but rather depends on the input data, and its tunable network parameters. Long Short-Term Memory (LSTM) recurrent networks were first introduced for sequential prediction tasks [12] and has since found successful applications in many areas [14], [15]. An LSTM layer consists of a set of recurrently connected blocks, known as memory blocks. As shown in Fig. 1, a basic block contains one memory cell with three multiplicative gates, which provide continuous analogues of write, read and reset operations for a cell in a digital memory chip. Specifically,

$$\begin{aligned} f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \\ i_t &= \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \\ C_t &= f_t * C_{t-1} + i_t * \widetilde{C}_t \\ o_t &= \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \\ h_t &= o_t * \widehat{C_t}, \end{aligned} \tag{3}$$

where W_f, W_i and W_o are weight matrices grouped with the corresponding gate, and σ is the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$. Cell state $\widetilde{C}_t$ and $\widehat{C_t}$ are candidate values that can be added to the cell state and output, both of them are computed through a tanh layer $g_{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$:

$$\widetilde{C_t} = g\left(W_c \cdot [h_{t-1}, x_t] + b_c\right), \quad \text{and} \quad \widehat{C_t} = g\left(C_t\right). \tag{4}$$

III. The Proposed Algorithm

We aim to use LSTM networks to learn the relatively longer-term dependencies of the filtering weight sequences based on the MCC-LMS filtering method. Fig. 2 shows the prediction context, which consists of pixels from the current band and three previous bands. Updating the weights according to Eq. (2) for each of the three co-located pixels from the previous bands will generate three separate sequences of weights, which are used to train LSTM networks, as shown in Fig. 3. The proposed prediction algorithm has the training stage and the prediction stage. In the training stage, context-based conditional average prediction (CCAP) [10] was first

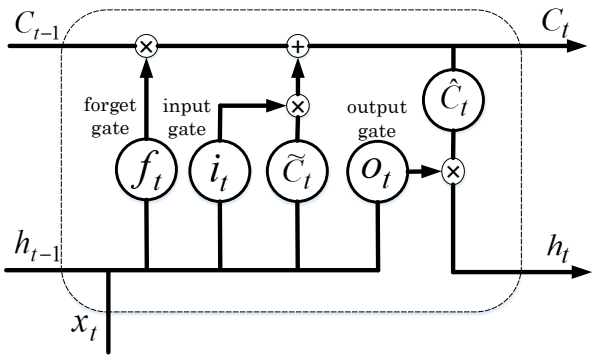


Fig. 1: A basic LSTM block consisting of a self-connected memory cell with three multiplicative gates – the input gate i_t, output gate o_t and forget gate f_t. The input data x_t and the output data from previous time step h_{t-1} are fed to each gate to determine the current cell state C_t, and the output h_t.

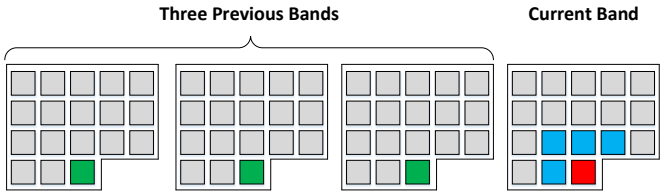


Fig. 2: Causal context (pixels in blue from the current band, as well as the pixels in green from the previous three bands) for prediction of the current pixel in red.

preformed on hyperspectral data to reduce the first-order entropy of the residual by taking the sample mean:

$$y_{i,j} = \frac{1}{|s(i,j)|}\sum_{(m,n)\in s(i,j)} x_{m,n}, \tag{5}$$

where $s(i,j)$ consists of four neighborhood pixels in the current band (colored blue in Fig. 2). The prediction stage has a closed-loop configuration by applying adaptive filtering twice, where the trained LSTM networks are used to regulate the filtering weights resulting from the first filtering operation, before the filtering is applied for the second time. Weights predicted by LSTM networks have taken into account the relatively longer term data dependencies. The weight updating formula at the n^{th} time instant for the second filtering operation can be written as:

$$W_{LSTM}(n) = LSTM(w(n)) \tag{6}$$

$$e_{LSTM}(n) = d_n - W_{LSTM}^T(n)X_n, \tag{7}$$

$$\begin{aligned} W_{LSTM}(n+1) =& W_{LSTM}(n) + \frac{\mu}{\sqrt{2\pi}\sigma^3} \\ & \exp\left(\frac{-{e_{LSTM}}^2(n)}{2\sigma^2}\right)e_{LSTM}(n)X_n, \end{aligned} \tag{8}$$

where $w(n)$ is the weight vector generated by adaptive filter, W_{LSTM} is the weight vector predicted by the LSTM network, and $e_{LSTM}(n)$ is the prediction error of the current pixel.

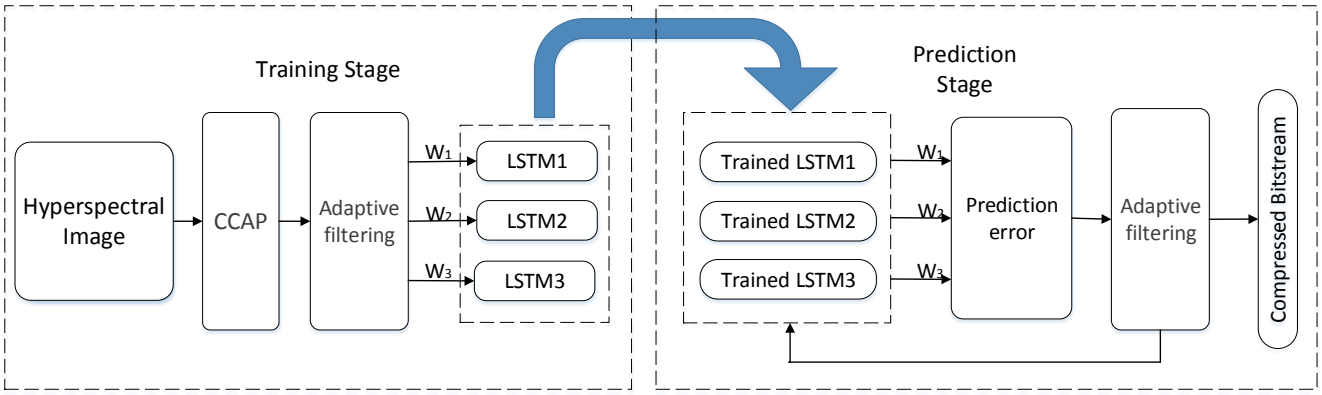


Fig. 3: Block diagram of the proposed method.

LSTM networks were trained by the weight sequences generated from the MCC-LMS filtering operations. Note that three individual LSTM blocks are assigned to three weight sequences, respectively. In particular, for each weight sequence, we constructed the training set and the corresponding labels by using the current weight value to predict the next one. That is, each element in the weight sequence can be viewed as a training data, and the label of the corresponding training data is the next weight value in the same sequence. The total size of the training set is thus closely related to the number of spectral bands. However, the time complexity would be an issue if we use this training method. By running MCC-LMS filtering, each spectral band involves three weight sequences, which means that we need to train three specific LSTM networks for each spectral band. It makes the filtering process extremely slow. We address this limitation by sharing the weights of LSTM networks across all the spectral bands. In other words, training each LSTM only once will be sufficient for prediction weights of all the bands. The reason we can do this is that the weight sequences from different bands tend to share similar dependency, since spectral bands are closely related to each other. Our simulations show that almost the same prediction errors are achieved by training and testing using the same spectral band versus a different band.

IV. Simulation Results

We tested the proposed algorithm on two hyperspectral image datasets: *Indian pines (IP)*, and *Pavia University (PU)* [16]. We experimented with different MCC-LMS parameters to achieve the best prediction results. As a result, we fix σ in Eq. (2) to 50 in our simulations. Since the number of bits per pixel varies from one dataset to another, we set the learning rate $\mu = 1$ for IP and $\mu = 0.1$ for PU. A small σ value will lead to a large learning rate and vice versa. Thus, a relatively large learning rate was selected for dataset with more bits per pixels to ensure faster convergence.

The LSTM blocks have one memory state for each block, since only one previous weight was used to predict the next one. We randomly selected a band for training LSTMs (band 6 was selected in our case), and the training set were built upon the weight sequences from band 6. Note that we only use 30 percent of the pixels in formulating training set for LSTMs (6307 entries for training each LSTM for *IP* and 62220 entries for training each LSTM for *PU*), thus only a small portion of the data was employed for training LSTMs compare to whole hyperspectral dataset. Data used for training LSTMs were excluded from testing. Note that while this architecture has been found to provide accurate prediction, it is not unique.

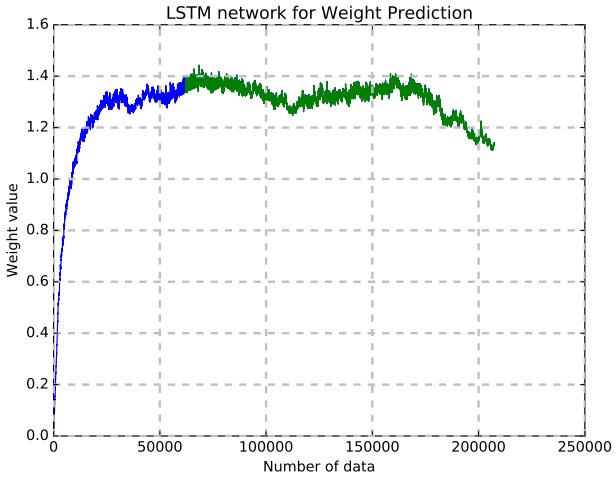


Fig. 4: Performance of LSTM network for weight prediction on *PU* dataset. All the weights were colored in blue, and the prediction results were colored in green.

Fig. 4 shows the weights prediction result of the LSTM network on a band of *PU* dataset. Note the weights were originally generated by adaptive filter, and first 30 percent of weights were used for training the LSTM network. It can be seen that the prediction result (in green) perfectly matches the real data (in blue). It means the LSTM network is competent in learning the long-term variation of weight sequence by only using a small portion of data. Thus, combining the LSTM with traditional adaptive filter leads to more precise pixel-value

prediction. We conducted additional simulations to measure the computational time in the training stage. The proposed method was coded based on Python 3.5 and Tensorflow 1.2.1 open source framework. The computational platform is a Thinkpad T460s with Intel Core i5-6300 (2.4GHz) and 8 GB RAM, running Windows 7 Professional (64-bit Operating System). No GPU acceleration was used for training. It takes 61.22 seconds in average to finish the training for each weight sequence from *IP* dataset. For relatively large *PU* dataset, it totally takes 71.45 seconds in average for each weight sequence.

We compare our algorithm with two existing adaptive filtering methods:

- The adaptive LMS method was proposed in [7] as the new CCSDS standard for hyerpsectral data compression. Adaptive LMS can be regarded as an LMS with adaptive learning rate. A simple CCAP is applied to the original dataset followed by conventional LMS-based filtering.
- MCC-LMS filtering based predictive compression was proposed in [6], which replaced the cost function of LMS with correntropy and achieved significant compression on regions of interest in hyperspectral images. Again, CCAP is applied to the original data first.

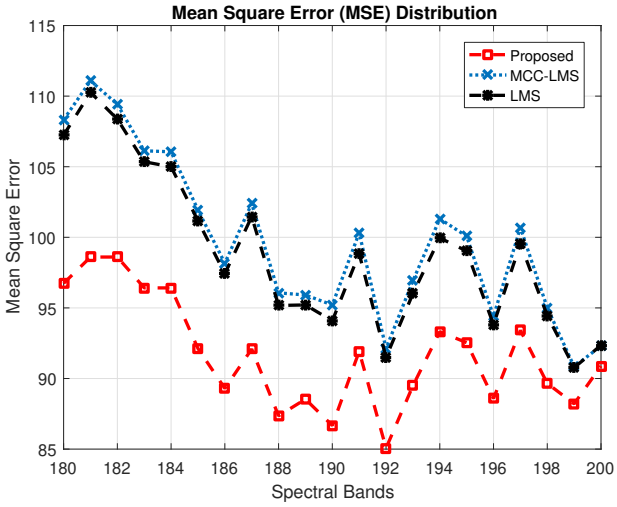
Note the residuals generated by the above two methods are encoded by a Golomb-Rice codes.

The prediction results of the three methods on two datasets show the performance improvement brought by using the proposed method. Table I shows the average prediction errors in RMSE (root-mean-square-error) of the three methods. It can be seen that our proposed method outperforms other two methods. This means that the LSTM networks are capable of capturing the dependencies of weight sequences, which can lead to more accurate pixel value prediction over the entire dataset.

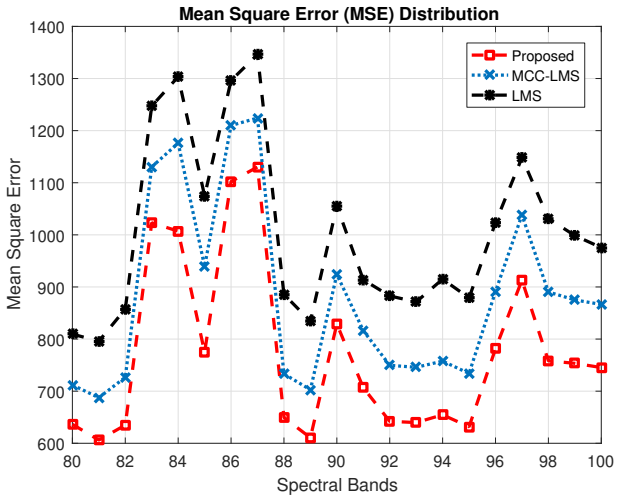
TABLE I: Comparison of Root Mean Square Error (RMSE) with two other methods on two datasets.

Dataset	LMS	MCC-LMS	Proposed
Indian Pines	110.8	105.7	**104.6**
Pavia University	47.4	46.3	**45.7**

Fig. 5 shows the band-by-band prediction errors (in MSE) of the three methods on two datasets. For the sake of displaying the evolution of the prediction errors clearly, we only show 21 spectral bands for each dataset. Apparently, the proposed method (red curves) achieved more accurate predictions than the other two methods consistently for all the spectral bands. In particular, Fig. 5(a) shows the prediction errors of the spectral bands ranging from 180 to 200, where the prediction errors are gradually reduced over the band. Fig. 5(b) shows how the prediction errors change for the PU dataset from spectral band 80 to 100. Again, the proposed method achieves the lowest prediction errors among all the bands. For example, the MSE of band 89 was decreased to 600, which is a significant improvement over other two methods. Note the variation of the prediction errors for the PU is larger than the IP dataset.



(a) Indian Pines (IP)



(b) Pavia University (PU)

Fig. 5: Average band-by-band prediction errors.

This might be caused by impulsive noises. The noise rejection property of LSTM network might have played an important role in smoothing the variations.

V. Conclusions and Future Work

We presented a novel adaptive filtering algorithm using LSTM network for hyperspectral images. LSTM networks appear to be effective in capturing the longer term dependencies of weight sequences. We proposed a two-stage framework by combining the trained LSTM networks with adaptive filters in a closed-loop configuration. Simulation results demonstrated we can reduce the prediction residuals, which would very likely lead to better compression. To the best of our knowledge, this is the first attempt to model not only the correlations between pixels from different spectral bands, but also the temporal dependencies of the filtering weights. We will evaluate the impact of reduced prediction errors on predictive lossless coding performance.

REFERENCES

[1] H. Shen, W. D. Pan, and Y. Wang, "A novel method for lossless compression of arbitrarily shaped regions of interest in hyperspectral imagery," in *Proc. IEEE SoutheastCon 2015*, Apr. 2015.

[2] E. Magli, "Multiband lossless compression of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 4, pp. 1168–1178, Apr. 2009.

[3] F. Rizzo, B. Carpentieri, G. Motta, and J. A. Storer, "High performance compression of hyperspectral imagery with reduced search complexity in the compressed domain," in *Proc. Data Compression Conf. 2004*, Mar. 2004, pp. 479–488.

[4] C. C. Lin and Y. T. Hwang, "An efficient lossless compression scheme for hyperspectral images using two-stage prediction," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 3, pp. 558–562, Jul. 2010.

[5] H. Shen, W. D. Pan, and D. Wu, "Predictive lossless compression of regions of interest in hyperspectral images with no-data regions," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, pp. 173–182, Jan. 2017.

[6] H. Shen and W. D. Pan, "Predictive lossless compression of regions of interest in hyperspectral image via maximum correntropy criterion based least mean square learning," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2016.

[7] M. Klimesh, "Low-complexity lossless compression of hyperspectral imagery via adaptive filtering," in *The Interplanetary Network Progress Report*, Jet Propulsion Laboratory, Pasadena, California, Nov. 2005, pp. 1–10.

[8] "Lossless multispectral & hyperspectral image compression CCSDS 123.0-B-1, ser.Blue Book, May 2012," https://public.ccsds.org/Pubs/123x0b1ec1.pdf, 2012 (accessed December 1, 2017).

[9] E. Magli, G. Olmo, and E. Quacchio, "Optimized onboard lossless and near-lossless compression of hyperspectral data using CALIC," *IEEE Trans. Geosci. Remote Sens.*, vol. 1, pp. 21–25, Jan. 2004.

[10] H. Wang, S. D. Babacan, and K. Sayood, "Lossless hyperspectral-image compression using context-based conditional average," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4187–4193, Dec. 2007.

[11] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, low-complexity image coding with a set partitioning embedded block coder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 11, pp. 1219–1235, Nov. 2004.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[13] W. Liu, P. P. Pokharel, and J. C. Principe, "Correntropy: properties and applications in non-Gaussian signal processing," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5286–5298, Nov. 2007.

[14] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. of the 31st Intl. Conf. on Machine Learning*, Jun. 2014, pp. 1764–1772.

[15] J. Y. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: deep networks for video classification," *arXiv preprint arXiv:1503.08909*, 2015.

[16] "Hyperspectral remote sensing scenes data," http://www.ehu.es/ccwintco/index.php?title=Hyperspectral Remote Sensing Scenes, accessed Aug. 30, 2014.