Hyperspectral data dimensionality reduction using nonlinear autoencoders

Evgeny Myasnikov
Geoinformatics and Information Security department
Samara National Research University;
Image Processing Systems Institute of RAS - Branch of the FSRC "Crystallography and Photonics" RAS
Samara, Russia
mevg@geosamara.ru

***Abstract*—The known feature of hyperspectral images is a high spectral resolution, which allows us to identify materials and classify objects in images with high accuracy. However hyperspectral images contain substantial redundancy, which can be eliminated with the aid of dimensionality reduction techniques. In this paper, we propose and study several dimensionality reduction techniques based on the pretraining the encoder-decoder neural network with the results of the nonlinear mapping and principal component analysis techniques. The experiments performed on an open dataset show that the proposed techniques both provide the discriminative low-dimensional features and allow us to reconstruct source hyperspectral data with little error.**

Keywords—autoencoder, hyperspectral images, nonlinear mapping, principal component analysis

I. Introduction

Hyperspectral images are widely used nowadays in different fields such as agriculture, medicine, biology, chemistry, and so on. The known feature of hyperspectral images is high spectral resolution, which allows us to identify materials and classify depicted images with high accuracy.

However hyperspectral images contain substantial redundancy, which can be eliminated with the aid of dimensionality reduction techniques. The images obtained after the dimensionality reduction stage can be processed efficiently as much less data volume is involved in processing. It is worth noting that dimensionality reduction techniques are often used in different problems of image analysis (see [1-3], for example). The key requirement to the dimensionality reduction procedures is the possibility to preserve the quality of the solution of applied problems that is classification, segmentation, material detection, and so on.

The most commonly used techniques for the dimensionality reduction of hyperspectral data are linear techniques such as Principal Component Analysis (PCA). While a number of general-purpose nonlinear dimensionality reduction procedures exist [4], their use in hyperspectral image analysis is limited as many of them do not provide the ability to restore source hyperspectral data as such procedures provide only one-way data mapping.

In the last years, neural network approaches become more and popular. In particular, autoencoder neural networks [5] were used for the dimensionality reduction of hyperspectral images. Such neural networks perform both nonlinear dimensionality reduction and provide the inverse mapping, which allows us to restore the source hyperspectral data up to some reconstruction error.

Recently, it was shown [6] that the autoencoder network can be pretrained using principal component analysis technique, and its use for the dimensionality reduction allowed to outperform the PCA technique both in terms of the reconstruction error and classification accuracy.

However, it was also shown [7,8] that the nonlinear mapping technique [9] have advantages over the PCA in terms of classification and segmentation quality of hyperspectral images. For this reason, in this paper, we study the possibility to train the autoencoder–like architecture to capture the nonlinear mapping. In particular, we split the autoencoder into encoder and decoder and train both parts separately using the results of nonlinear mapping and investigate the effect of the subsequent fine-tuning of the whole network.

The structure of the paper is as follows. In the next Section II, we give necessary theoretical information on the neural network architecture and the nonlinear mapping algorithm. In Section III we describe the training procedures used in the experimental study and describe the results of experiments. The conclusions and the list of references are given at the end of the paper.

II. Method

A. Autoencoder Neural Network

The autoencoder neural network proposed in [5] was earlier referred to as the autoassociative neural network. It consists of two consecutive parts called the encoder and decoder.

The encoder part takes a multidimensional vector $x \epsilon R^M$ as input and produces corresponding low-dimensional representations $y \epsilon R^m$ so that $m<M$. The encoder consists of at least two fully – connected layers. The first layer contains some number of neurons (defined by the parameters of the neural network architecture) connected to all the components of an input vector. The last layer of the encoder contains the number of neurons equal to the desired dimensionality of the reduced space.

The decoder usually has the mirror-reflected architecture. It has the same number of layers with the same number of neurons, but this is not the necessary requirement. Anyway, the input layer of the decoder takes the reduced representation $y \epsilon R^m$ from the output of the encoder and restores the multidimensional vectors $\tilde{x} \epsilon R^M$. So the output layer of the decoder has the number of neurons equal to the input dimensionality M. The number of hidden layers and neurons is defined by the parameters of the neural network architecture.

As the number of neurons in the output layer of the encoder is less than the number of neurons in the input and hidden layers, this layer is often referred to as a bottleneck layer, and the whole network architecture is often referred to as a bottleneck architecture.

The autoencoder architecture is usually trained in self-learning mode by applying the same multidimensional vectors $x \epsilon R^M$ to both input and output layers of the autoencoder. The training process itself is based on the minimization of the following cast function:

$$E = \frac{1}{N} \sum_{i=1}^{N} \left\| x_i - \tilde{x}_i \right\|^2 \qquad (1)$$

where N is the number of samples, and $x \epsilon R^M$, $\tilde{x} \epsilon R^M$ are inputs and outputs of the network. After training the encoder can be used to perform the dimensionality reduction of the source data (direct mapping), and decoder can be used to restore the source data by its reduced representation (inverse mapping).

In this paper, we study, if the encoder and decoder parts can be trained separately to force the neural network to perform the mapping with the desired properties. It was shown earlier that the separate pre-training of encoder and decoder with the PCA results helped to perform the training more efficiently compared to the standard training.

In particular, the approach proposed in [6] consists of the following steps: perform the PCA for the input dataset; pre-train the encoder to produce the PCA results for the input data; pre-train the decoder to produce the input data for the encoded data; fine-tune the whole network according to the standard scheme.

In this paper, we follow the similar scheme but use the results of the nonlinear mapping algorithm instead of the PCA, and perform the fine-tuning optionally to study if such an approach can be more efficient than the standard PCA, nonlinear mapping or the proposed recently autoencoder pretrained with the PCA [6].

B. *Nonlinear Mapping*

The nonlinear mapping is a numerical procedure that performs the mapping (nonfunctional) of data into low-dimensional space so that the data structure is preserved (see [8] for example). This structure is defined in nonlinear mapping by all the pairwise distances between the points in the dataset. The Euclidean distance *d()* is usually used to measure the distances.

As the pairwise distances cannot be preserved exactly in a common case, the so-called data mapping error is introduced:

$$\varepsilon = \mu \cdot \sum_{i,j=1\,(i<j)}^{N} \left(\rho_{i,j} \left(d(x_i, x_j) - d(y_i, y_j) \right)^2 \right) \qquad (2)$$

Here N is the number of data points, $d(x_i,x_j)$ is the distance between points x_i and x_j in the multidimensional space, $d(y_i,y_j)$ is the distance between the corresponding points y_i, y_j in the reduced space, μ and ρ are some constants. Usually, μ is the inversion of the sum of square distances between all the possible pairs of data points in multidimensional space, and ρ_{ij} are equal to one.

The minimization of the data mapping error is usually performed using the gradient descent technique. The coordinates of data points $y_i \epsilon R^m$ are the tunable parameters.

In this paper, we use the stochastic gradient descent based on mini-batches to minimize the data mapping error. The overall algorithm for dimensionality reduction using the nonlinear mapping consists of the initialization of the coordinates y_i with the results of the principal component analysis with the subsequent refinement of y_i using the stochastic gradient descent. The optimization process (refinement) stops when the coordinates of the data points y_i in the reduced space become stable.

C. *The methods used in the study*

As it was outlined in the introduction, in this paper, we study several variants of training the autoencoder-like encoder-decoder network. In particular, we consider the following techniques:

- The autoencoder network pretrained with the results of the PCA technique (AE-PCA), as it is described in [6];

- The neural network with encoder and decoder (ED-NLM) trained separately using the results of the nonlinear mapping technique;

- The same autoencoder network pretrained with the results of the nonlinear mapping technique and fine-tuned using the standard approach (AE-NLM).

III. EXPERIMENTS

In this section, we describe the results of the experiments, which were performed using the Indian Pines dataset. This dataset was acquired using the AVIRIS hyperspectral sensor. This dataset contains 145 x 145 image pixels and 224 spectral components [10]. Due to the high noise and water absorption in the source image, we used the version containing 204 spectral channels.

In all the described experiments, for the implementation of the neural networks, we used the Keras framework and Python language. The experiments were carried out on GeForce GTX 1070 ti.

For each considered neural network technique, we varied the number of hidden layers in the encoder and decoder and performed experiments for one and two hidden layers that correspond to four and six layers in the corresponding autoencoder networks.

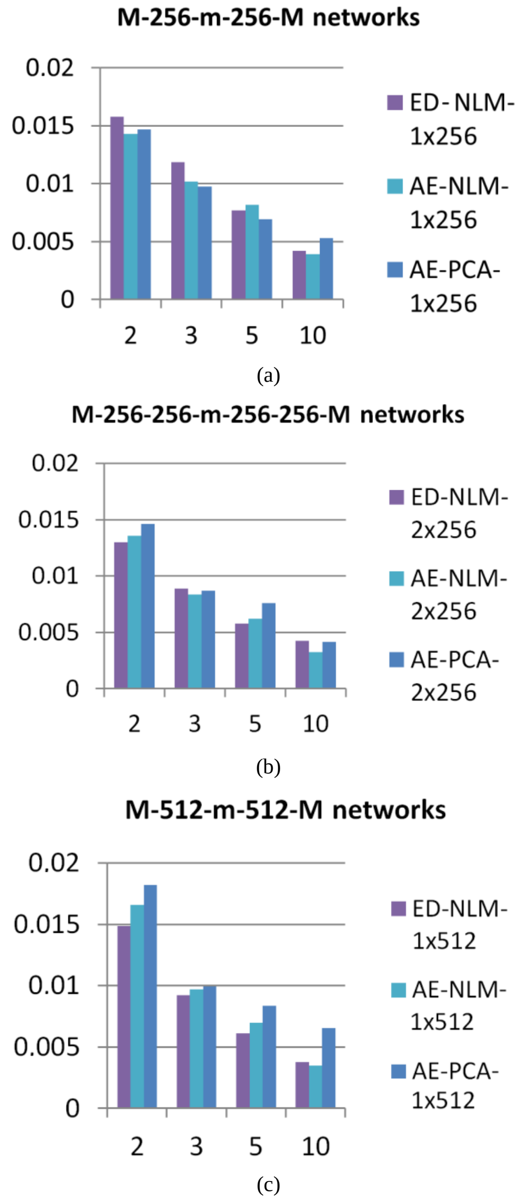
The number of neurons in the input layer of the encoder and the output layer of the decoder was defined by the dimensionality of the input space that is the number of channels in the hyperspectral image. The number of neurons in the bottleneck layer varied from 1 to 10 according to the dimensionality of the reduced space. We also varied the number of neurons in the hidden layers. In particular, we used 64, 128, and 256 neurons in hidden layers.

According to the recommendations given in [6], we used ReLU activation functions for hidden layers and linear activations in the output layers of the encoder and decoder. Analogously, we used Adam optimizer [11] with the default parameters. The batch size was set to 16, however, we suppose that a bigger batch size could also be used.

To measure the effectiveness of each particular approach, we estimated both the reconstruction error as it is defined in (1) and the classification accuracy using the reduced representation. The latter indicator plays an important role in hyperspectral image analysis problems, for example, in vegetation type recognition [12].

For the latter indicator, we used the overall accuracy of the one nearest neighbor (1-NN) classifier. The accuracy itself was measured as a fraction of correctly classified image pixels. To measure the accuracy, at first, we performed dimensionality reduction using one of the studied techniques for all the pixels in the considered image. Then we split all the ground truth pixels into training and testing sets in the proportion 60/40. After that, we trained the classifier using the training set and estimated its accuracy using the test one.

In our first experiment, we compared different techniques described in Subsection II.C and different architectures from the viewpoint of the reconstruction error (1). The results of this experiment are shown in Fig. 1. In particular, we pre-trained the encoder and decoder of the AE-PCA network for 50 iterations, fine-tuned the entire network for 50 iterations, and then measures the reconstruction quality.



(a)

(b)

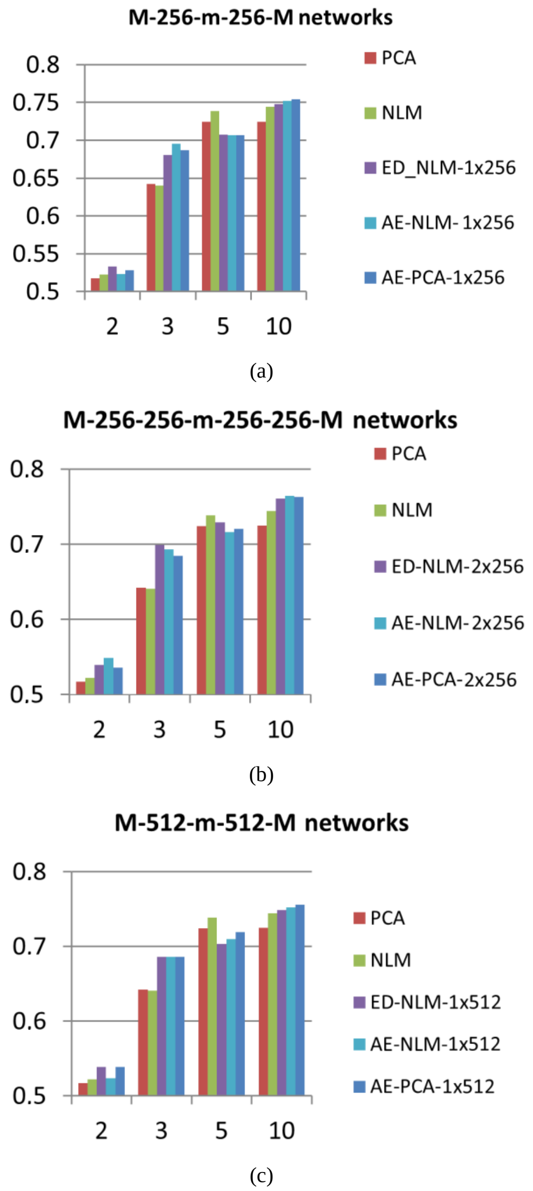
(c)

Fig. 1. The reconstruction error for different techniques and network architectures (a-c).

For the AE-NLM network, we trained the network with the same strategy, but used the NLM results instead of the PCA results at the pretraining stage. For the ED-NLM network, we trained separately encoder and decoder for 100 epochs. After the training, we measured the error (1) as the quality indicator. The experiment was carried out for a different number of layers and neurons.

As can be seen in the figure, the reconstruction error decreases with the growth of the dimensionality *m* of the reduced space defined by the number of neurons in the bottleneck layer, which is an expected result.

While we cannot highlight any winner technique in this experiment, we should note, that the AE-NLM technique often shows better results. It means that the nonlinear mapping result, which was used for training, provide the ability to restore the source data with quite a good quality. This also means that the decoder trained on the NLM data can be used as an inverse mapping for the NLM.



(a)

(b)

(c)

Fig. 2. The classification accuracy for different techniques and network architectures (a-c).

In our second experiment, we compared the considered techniques from the viewpoint of the classification accuracy. The results of this experiment are shown in Fig. 2. In this figure, we added the results for the classical linear (PCA) and nonlinear (NLM) dimensionality reduction techniques.

As can be seen, the proposed techniques provided better results than the classical approaches in most cases. Again, it is difficult to outline any approach. Nevertheless, we do not observe any substantial advantages in the fine-tuning of the NLM initialized network over the version with separate encoder and decoder.

IV. CONCLUSION

In this paper, we studied several dimensionality reduction neural network techniques based on autoencoder architecture. We compared the proposed techniques from the viewpoint of the reconstruction error and the accuracy of the per-pixel classification.

We showed that the proposed techniques outperformed the baseline (PCA and NLM) approached in terms of the classification accuracy in almost all the considered cases. The decoder trained using the results of the NLM can be successfully used as an inverse mapping for hyperspectral image analysis.

ACKNOWLEDGMENT

The work was partly funded by RFBR according to the research project 18-07-01312-a in parts of «2. Method» - «3. Experiments» and by the Russian Federation Ministry of Science and Higher Education within a state contract with the «Crystallography and Photonics» Research Center of the RAS in parts «1. Introduction» and «4. Conclusion».

REFERENCES

[1] E.A. Dmitriev and V.V. Myasnikov, “Comparative study of description algorithms for complex-valued gradient fields of digital images using linear dimensionality reduction methods,” Computer Optics, vol. 42, no. 5, pp. 822-828, 2018. DOI: 10.18287/2412-6179-2018-42-5-822-828.

[2] M.V. Gashnikov, “Optimization of the multidimensional signal interpolator in a lower dimensional space,” Computer Optics, vol. 43, no. 4, pp. 653-660, 2019. DOI: 10.18287/2412-6179-2019-43-4-653-660.

[3] E.V. Myasnikov, “The study of dimensionality reduction methods in the task of browsing of digital image collections,” Computer Optics, vol. 32, no. 3, pp. 296-301, 2008.

[4] J.A. Lee and M. Verleysen, “Nonlinear Dimensionality Reduction," Springer, 2007.

[5] M.A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” AIChE J., vol. 37, pp. 233-243, 1991.

[6] E. Myasnikov, “Dimensionality Reduction of Hyperspectral Images using Autoassociative Neural Networks,” IEEE Proc. of International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), pp. 0591-0595, 2019.

[7] E. Myasnikov, “Evaluation of nonlinear dimensionality reduction techniques for classification of hyperspectral images,” CEUR Workshop Proceedings, vol. 2268, pp. 147-154, 2018.

[8] S. A. Bibikov, N. L. Kazanskiy and V. A. Fursov, “Vegetation type recognition in hyperspectral images using a conjugacy indicator,” Computer Optics, vol. 42, no. 5, pp. 846-854, 2018. DOI: 10.18287/2412-6179-2018-42-5-846-854.

[9] J.W. Sammon, “A nonlinear mapping for data structure analysis,” IEEE Transactions on Computers, vol. 18, no. 5, pp. 401-409, 1969.

[10] M.F. Baumgardner, L.L. Biehl and D.A. Landgrebe, “220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3,” Purdue University Research Repository, 2015. DOI:10.4231/R7RX991C.

[11] D. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” arXiv: 1412.6980v8, 2017.

[12] S.A. Bibikov, N.L. Kazanskiy and V.A. Fursov, “Vegetation type recognition in hyperspectral images using a conjugacy indicator,” Computer Optics, vol. 42, no. 5, pp. 846-854, 2018. DOI: 10.18287/2412-6179-2018-42-5-846-854.