Hyperspectral image classification using deep convolutional neural networks

Zilong Zhong University of Waterloo, ON, Canada

Jonathan Li University of Waterloo, ON, Canada

Abstract

The prevailing framework consisted of complex feature extractors following by conventional classifiers. Nevertheless, the high spatial and high spectral dimensionality of each pixel in the hyperspectral imagery hinders the development of hyperspectral image classification. Fortunately, since 2012, deep learning models, which can extract the hierarchical features of large amounts of daily three-channel optical images, have emerged as a better alternative to their shallow learning counterparts. Within all deep learning models, convolutional neural networks (CNNs) exhibit convincing and stunning ability to process a huge mass of data. In this paper, the CNNs have been adopted as an end-to-end pixelwise scheme to classify the pixels of hyperspectral imagery, in which each pixel contains hundreds of continuous spectral bands. According to the preliminarily qualitative and quantitative results, the existing CNN models achieve promising classification accuracy and process effectively and robustly on the University of Pavia dataset.

1. Introduction

With the increasing number of airborne and space borne sensors, large amounts of high spatial, high spectral and high temporal resolution remotely sensed data are available for multiple kinds of scientific, military and civilian purposes [1-2]. An incomplete list consists of agricultural monitoring, security surveillance, disaster management, urban planning, land cover and land use analysis, and climate change surveying. Among all types of remote sensing data processing, hyperspectral imagery (HSI) classification, which means labeling each pixel of hyperspectral imagery with a certain type of land cover, attracts a lot of attention from different academic fields and becomes an interdisciplinary study.

In recent years, deep learning models – which include deep belief network (DBN) [3], auto-encoder (AE) [4], and convolutional neural network (CNN) – have achieved multiple times the highest accuracy in the challenging contests of image classification, image segmentation, object detection, speech recognition and thematic labeling [4-6]. In 2006, Hinton et al. revitalized the tremendous enthusiasm of deep neural networks in computer vision and pattern recognition domains [7]. Equally important in 2012, Krizhevsky et al. demonstrated the powerful feature learning ability of deep CNNs in a large-scale labeled-image classification contest [8]. At that time, because the shortage of powerful hardware graphic processing units (GPUs), the model was implemented in two GPUs parallelly. Concurrently, Mnih applied the CNN models to labeling buildings and streets using high spatial resolution imagery and achieved promising results [9].

Some articles have tried to incorporate the deep learning models into the interpretation task of hyperspectral images. In 2014, Chen et al. tested the deep feature learning ability of AE, which is the first deep learning model that has been used in this task, in two real hyperspectral datasets [10]. Recently, some papers have tried to use CNN models to classify HSI, but did not fully explain the multilevel feature learning ability of CNN [11]. We adopted existing and newly designed CNN models as a pixelwise spectral classifier to test their characteristics and conduct experiments using the open source hyperspectral imagery of the urban scene.

This paper mainly focus on three aspects. First, we estimate the performance of two popular existing deep CNN models for traditional spectral information interpretation. Second, we analyze the basic fabric of CNN models and test models of different layers on an urban hyperspectral dataset. Finally, we discuss the potential approaches that can further improve the classification accuracy. For example, conditional random fields can be incorporated as a regularization procedure that stresses the prior spatial-contextual information conditioned on the classification outputs from the previous steps.

2. Convolutional Neural Networks

The discriminative properties of CNN models that distinct from other deep learning models mainly lies in three perspectives: (1) Convolutional layers and pooling layers inherently stress the importance of both spatial and contextual information and contribute to the reduction of dimensionality of data space; (2) Local receptive fields that guarantee the sparsity of the learned feature space;(3) Deep structure that helps the model easily learning the hierarchical and abstract semantic information. Although other deep learning models also could have very deep layers, the CNN models have an incomparable powerful performance regarding the learning speed and the ability to handling high dimensional data.

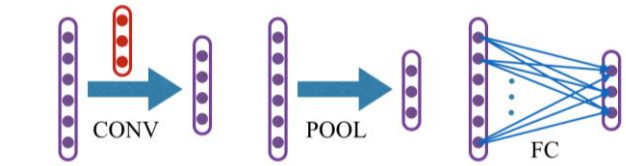

Fig. 1. Basic layers of CNNs

2.1 Basic structure of CNNs

Fig. 1 shows three basic operational layers in CNNs: convolutional layer, pooling layer, and fully connected layer. It is obviously that the prominent structure of CNN models is the convolutional layers, which employ a bank of filters to extract features in a hierarchical way and share weights with the same corresponding neuron.

Pooling layers also contribute a lot to reduce the dimensionality and thus facilitate the training process. Both average and maximum pooling could generate reasonable results, we select the maximum pooling in this project. The fully connected layers are nothing but the same as general neural networks. Although the fully connected layers contain the least

number of neurons, they contribute most of the training time during the forward inference and backward learning iterations.

Recently, a lot of new activation function have been designed, some of which can increase the classification performance a bit. However, the newly designed activation function cannot generate fundamental improvement. Therefore, we still employ the most commonly used rectified linear unit (ReLU) as the activation function.

$$f(x) = \max(0, x) \quad (1)$$

As a generally supervised machine learning procedure, we need to build an object function, as known as loss function, to estimate the distance between the output prediction and the ground truth labeling and to configure the process of training. The loss function adopted in this project is the multinomial logistic regression.

2.2 LeNet and AlexNet

The LeNet was designed for hand written numbers. Owing to the introduction of convolution layers and the lightness of its framework, LeNet achieved a huge success in the early 1990s and is, in fact, the preliminary version of the models, including the AlexNet, that emerged afterward. The LeNet consists of three convolutional layers, two maximum pooling layers, and three fully connected layers. On the contrary, the AlexNet was built for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) held in 2012 and obtained the best results within all submitted models [8]. The AlexNet is composed of five convolutional layers, three pooling layers, and three fully connected layers.

After setting the model, a training strategy, as known as a solver, should be selected to control the training process. We choose Nesterov's accelerated gradient (NAG), which makes the training process faster than the models using stochastic gradient descent (SGD) as the solver. Besides, several hyper-parameters, such as learning rate and training epoch, also have a great influence on the training process If the learning rate is set to be too high, the training process is not likely to converge and the value of loss function always stays high. In contrast, if the learning rate is too small, the training process will become slow and the trained model is easily stuck into the local minimum of the solution space.

3. Experiment results

Caffe [12], which provides abundant resources and application interfaces for the development of CNNs, is the framework We used to train and test models. To estimate the two off-the-shelf CNN models and test the basic layers of CNNs, two experiments have been designed in this project. First, as the shortage of labeled data, Monte Carlo sampling method has been adopted to demonstrate the effectiveness and robustness of the trained models. We conduct 10 independent times of the whole training and testing processes on 10 different distribution of the UPavia dataset. Second, for demonstrating the property of the convolutional layer, three CNN models including one convolutional layer, two convolutional layers, and three convolutional layers are built and tested using the same hyperspectral dataset. Overall accuracy, average accuracy, and kappa coefficients are used to assess different models. In addition, the classification results of all the trained models are also presented as qualitative evaluation.

3.1 Preprocess

Considering the deep learning models need large amounts of data to fully unfold their capability, we choose the widely-used University of Pavia (UPavia) dataset, which has a very high resolution and enough number of samples to train CNN models. The UPavia dataset, which was acquired by the reflective optics system imaging spectrometer (ROSIS) sensor during a flight over northern Italy, contains 610×340 pixels of 103 bands with the 1.3-meter resolution. All the hyperspectral pixels are classified into nine categories.

According to [13], the whole dataset of UPavia has been separated into three groups. 60%, 20%, and 20% of the whole hyperspectral pixels have been deployed into training, validation and testing datasets. When separating the data, we transform the 103-dimensional vector of each pixel into a corresponding image with the size of 103×103 through simple duplication.

3.2 Training process monitoring

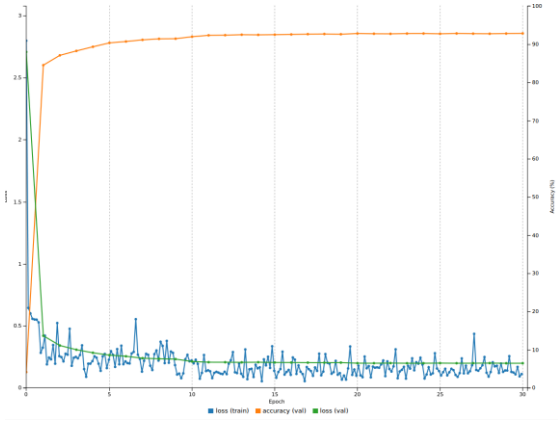

Fig. 2. The changing of training loss, accuracy and validation loss of LeNet

The initial learning rate was set to be 0.001 for LeNet and 0.01 for AlexNet. Moreover, for conducting more experiments, the training epoch is set to be 30 for all trained CNN models. The deeper the models are, the smaller the initial learning rate could set. The changing tendency regarding the loss value for both training and validation data is demonstrated in Fig. 2. Apparently, the accuracy is rising with the training epoch increasing and the decreasing of learning rate. Accordingly, the values of both the training and validation loss appear to have the opposite trend.

3.3 LeNet and AlexNet

With 10 minutes, the newly trained LeNet model achieved reasonable accuracy (OV = 91.71%, AA = 89.39%, and Kappa = 0.8892) for pixelwise classification after only 30 epochs training. Specifically, this trained model performs very well for the Meadows class. However, for the Bitumen and Metal sheets categories, the LeNet model did not obtain high classification accuracy. To further test the credibility of the results, we conduct the same experiment for nine more times using independent sampling data, with which the LeNet and AlexNet models trained respectively.

After 10 times of independent sampling, for the trained LeNet model, the mean overall accuracy, average accuracy and Kappa coefficient are 93.11%, 91.24% and 0.9083. The average classification accuracy for LeNet is a little bit higher than that of AlexNet. However, from the standard deviation point of view, the AlexNet performed more robust than LeNet. My explanation of this phenomenon lies in the model scale of the two network. On the one hand, the AlexNet has larger network size than LeNet and the large size benefit the robustness. On the other hand, the relatively small LeNet has lower robustness, but the scale of the model fits better than that of AlexNet.

3.4 Designed CNN models

We tested three different layers of CNNs for comparison. The Conv_1 model consists of a convolutional layer followed by a pooling layer. Besides, the Conv_2 model contains two convolutional layers without pooling layers and the Conv_3 model is composed of three convolutional layers. To compare all aforementioned models, the training and testing data of the first sampling process is adopted as the benchmark. As shown in Fig. 4, the overall accuracy is given in the parenthoods for each model and the AlexNet obtained the high overall accuracy among all five trained models.

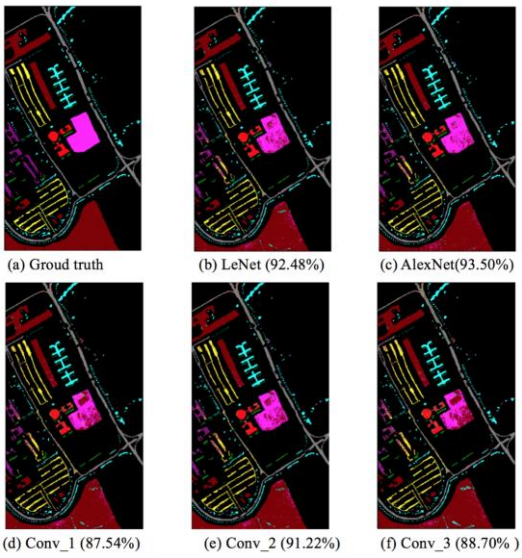

Fig. 4. The output of CNN models trained and tested on the same dataset

As to the three-new trained model presented by Fig. 5. (d-e), the Conv_2 model outperformed the other two with apparent superiority, which implies the importance of a moderate scale regarding given datasets. The simplest Con_1 model still achieved the overall accuracy of 87.54%. The overall accuracy of Conv_2 is higher than that of the Conv_1 model shows that the convolutional layer has better feature learning methods than the pooling layer. However, with the layer goes deeper, the performance of CNN model deteriorates. So, it is of vital importance to decide the best number of layers for classification using given dataset.

4. Conclusions

This project presents two off-the-shelf deep convolutional models of 2-D optical images that can be directly employed in hyperspectral image classification task. Moreover, three new models are designed to evaluate the classification performance of simple CNN models and presenting the ability of the basic structure of CNN models. According to the quantitative and qualitative results, the two experiments demonstrate the promising and reasonable classification performance on the urban hyperspectral dataset as pixelwise classifiers.

As to the future research steps for this project, we have three major thoughts: First, combination of spatial information. Most the state-of-the-art classification models are spatial-spectral classifiers, which incorporated the strengths of both spectral and spatial-domain classifiers, that considered contextual prior information and pixelwise spectral signature. Second, comparing with the state-of-art. [10] describes the process of network designing, training and contrast experiments using stacked AE. Similarly, we will mainly focus on the implementation of the algorithm and comparison experiments using different types of hyperspectral. Finally, extensive experiments for selecting the best number of layers and hidden layer size.

References

[1] Bioucas-Dias, J.M., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N.M. and Chanussot, J., 2013. Hyperspectral remote sensing data analysis and future challenges. *Geoscience and Remote Sensing Magazine, IEEE*, *1*(2), pp.6-36.

[2] Z. Zhong, J. Li, W. Cui, and H. Jiang, "Fully convolutional networks for building and road extraction: Preliminary results," in *proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 1591-1594.

[3] Hinton, G.E., Osindero, S. and Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. Neural computation, 18(7), pp.1527-1554.

[4] Glorot, X., Bordes, A. and Bengio, Y., 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (pp. 513-520).

[5] Salakhutdinov, R. and Hinton, G.E., 2009. Deep boltzmann machines. In*International conference* on artificial intelligence and statistics (pp. 448-455).

[6] Hinton, G.E. and Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), pp.504-507.

[7] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *Nature*, *521*(7553), pp.436-444.

[8] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[9] Mnih, V., 2013. Machine learning for aerial image labeling, PhD dissertation, University of Toronto, 109p.

[10] Chen, Y., Lin, Z., Zhao, X., Wang, G. and Gu, Y., 2014. Deep learning-based classification of hyperspectral data. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, *7*(6), pp.2094-2107.

[11] Liang, H. and Li, Q., 2016. Hyperspectral Imagery Classification Using Sparse Representations of Convolutional Neural Network Features. *Remote Sensing*, *8*(2), p.99.

[12] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), pp.2278-2324.

[13] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T., 2014, November. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia* (pp. 675-678). ACM.