Hyperspectral image classification based on different fine-tuning methods and HybridSN

Zhang JiangHao
Chongqing University
Chongqing, China
20183917@cqu.edu.cn

Abstract

Deep learning has become a hot topic in the field of Hyperspectral image (HSI) classification. Spectral imaging divides the spectrum into varying bands and it hinders existing model from recognizing real world hyperspectral images. Transfer Learning is a decent method to add generalization ability of existing HSI model and it can help to reducce consumption of computational units; And fine-tuning is the most basic deep transfer learning method. In order to overcome the limitation of poor generalization ability and reduce consumption of computational units, this paper implements and compares seven various methods of fine-tuning on the basis of the existing models to implement image semantic segmentation of new HSI data set like Urban data set.

1. Introduction

Hyperspectral image classification plays an important role in remote sensing technology, which collects electromagnetic spectrum from visible to near-infrared bands. In view of the advantages and challenges of spectral information in the identification of material attributes, hyperspectral remote sensing has become the forefront of the development of international remote sensing technology. Recently, deep learning has become a main trend in semantic image segmentation and has already overcome countless vision problems, motivated by those successful applications, deep learning has been implemented to recognize hyperspectral images (HSIs), and achieved good performance [6].

However, due to the influence of various factors such as illumination, environment and climate, the real hyperspectral data are often quite different from the available data sets.In order to reduce the negative impact of this difference on hyperspectral image recognition, we need to improve the generalization of deep learning model, and Transfer learning, where the goal is to Transfer knowledge from a related source task,is commonly used to compensate for the lack of sufficient training data in the target task [7]. Therefore, it is of great significance to apply transfer learning to field of HSI.

Fine-tuning is arguably the most widely used approach for transfer learning when working with deep learning models. It starts with a pre-trained model on the source task. Compared with training from original state, implement fine-tuning to a pre-trained convolutional neural network (CNN) on a target dataset can dramatically improve performance, while reducing the target labeled data requirements [4].

In this paper, I tested 7 different fine-tuning methods on a new data set -- Urban data set on the basis of a trained model called Hybrid Spectral CNN (HybridSN) [8] based on Indian Pines (IP) data set, and finally found that two of them have far higher performance; The highest accuracy on the test set using these two methods are basically the same as the direct training method, reaching about 80%, but the calculation units and training time used in such training process are greatly reduced, which verifies the generali- zation ability of HybridSN and the importance of transfer learning in the field of hyperspectral image recognition to a certain extent.

2. Related Work

Hyperspectral image (HSI) classification is a phenomenal mechanism to analyze diversified land cover in remotely sensed hyperspectral images [2]. In the early stage of the study on HSI classification, most methods have focused on exploring the role of the spectral and some other classification approaches have focused on designing an effective feature-extraction or dimension-reduction technique, such as principal component analysis (PCA) [6]. However, the classification maps obtained by these pixel-wise classifiers are unsatisfactory since the spatial contexts are not considered. Recently, motivited by the out-standing performance of applications of deep learning, such methods are introduced into the field of HSI [8].

Transfer Learning (TL) Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. [5]. From the practical standpoint, reusing or transferring information from previously learned tasks for the learning of new tasks has the potential to significantly improve the sample efficiency of a reinforcement learning agent. In recent years, TL has been successfully applied in many fields, especially in the remote sensing field [6], where the available training samples are not sufficient.

3. Proposed Approach

In order to find the best method of fine–tuning, expand robustness and generalization of the existing HSI model, and to reduce the number of computing units while training, I defined 7 methods of fine–tuning by a pre-trained network model with different types and number of frozen layers. After the model is trained on another hyperspectral data set, we can achieve our goal by comparing the results of the model without pre-training by the convergence rate of the model, the highest accuracy of the test set and the number of computing units consumed.

3.1. Introduction of Hybrid Spectral CNN (HybridSN)

Table 1. Layerwise Summary of the Original Hybridsn Architecture. the Last Layer is Based on the Ip Data Set.

Layer (type)	Output Shape	# Parameter
Input_1 (InputLayer)	(25, 25, 30, 1)	0
Conv3d_1 (Conv3D)	(23, 23, 24, 8)	512
Conv3d_2 (Conv3D)	(21, 21, 20, 16)	5776
Conv3d_3 (Conv3D)	(19, 19, 18, 32)	13856
Reshape_1 (Reshape)	(19, 19, 576)	0
Conv2d_1 (Conv2D)	(17, 17, 64)	331840
Flatten_1 (Flatten)	(18496)	0
Dense_1 (Dense)	(256)	4735232
Dropout_1 (Dropout)	(256)	0
Dense_2 (Dense)	(128)	32896
Dropout_2 (Dropout)	(128)	0
Dense_3 (Dense)	(16)	2064
Total Trainable Parameters: 5, 122, 176		

HybridSN was a convolutional neural network model for semantic segmentation of HSIs proposed by Dubey [8] in 2019 and it was the state of the art on the test of IP data set. Its remarkable performance comes from combination of 3-D convolution and 2-D convolution: The 3-D convolution facilitates the joint spatial–spectral feature representation from a stack of spectral bands and the followed 2-D convolutional not only reduces the complexity of network but also learns more abstract feature-level features, just like a bottleneck layer. The whole original architecture is shown in TABLE 1, the following work is based on this model.

3.2. Fine-tuning & Theoretical Guarantee of TL

In practical applications, we usually do not train a neural network from scratch for a new task. Such operation is ob -viously time consuming. In particular, our training data is unlikely to be as large as ImageNet and is often not able to build deep neural networks that are strong enough to generalize. Transfer learning indicates that we can utilize pre-trained models to tansfer to our target domain.

In such pre-trained models, the total kinds of output doesn't always fit target ouput, so we have to modify or fine-tune some of mismatching layers. The following picture shows a simplest fine-tuning method:

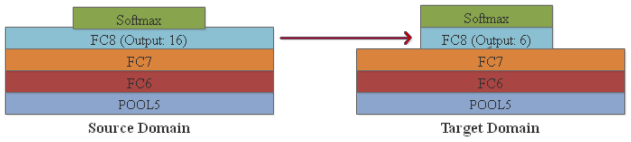

Figure 1. Illustration of a simple situation of transfer learning that has to apply fine-tuning.

In terms of current research results, theoretical work in the field of transfer learning is very lacking. David's work [1] answers the question that within what margin of error is it feasible to learn from different areas:

Learning error: giving two domains D_s & D_t, X is the data defined over them, and the divergence of class H between these two domains can be defined as Equation (1):

$$\hat{d}_H(D_s, D_t) = 2\sup_{\eta\in H}\left|\underset{x\in D_s}{P}[\eta(x)=1] - \underset{x\in D_t}{P}[\eta(x)=1]\right| \tag{1}$$

As a result, the divergence relys on the where the data comes from and David proved that for every symmetric H, the divergence can calculated by following Equation (2):

$$d_H(D_s, D_t) - 2\left(1 - \min_{\eta\in H}\left[\frac{1}{n_1}\sum_{i=1}^{n_1} I[\eta(x_i)=0] + \frac{1}{n_2}\sum_{i=1}^{n_2} I[\eta(x_i)=1]\right]\right) \tag{2}$$

Where $I[a]$ is the indication function: $I[a] = 1$ when a is established.

3.3. Different Kinds of Fine-tuning Methods

I defined different kinds of fine-tuning methods by freezing diverse numbers or types of layers. And I named these HybridSN-based transfer learning model by the kind of fine-tuning methods as following TABLE II and an example is avaible in Figure 2:

Figure 2:

Table 2. Network Defined by Differenr Types of Fine-tuning Methods & Corresponding Trainable Layers (not Frozen Layers)

Network Name	Trainable Layers
Network_Orginal	all
Network_1	Dense_3, Added Layer
Network_2	Dense_3
Network_3	Dense_3, Dense_2
Network_4	Dense_3, Dense_2, Dense_1
Network_5	Dense_3, Dense_2, Dense_1, Conv2d_1
Network_6	Dense_3, Conv2d_1
Network_7	Dense_3, Dense_2, Dense_1, Conv2d_1, Conv3d_3

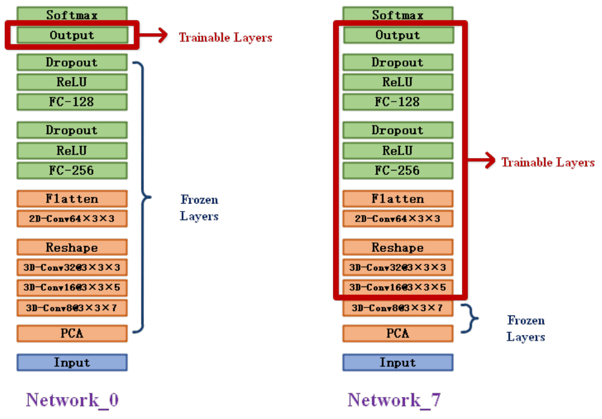

Figure 2. Illustration of two network architecture mentioned in TABLE II.

4. Experiments

4.1. Data set Description and Training Details:

I used two publicly HSI data set: one is Indian Pines (IP) and another is Urban_F210 data set which was a new HSI data set and hadn't been applied in HybridSN. The images of IP data set has 145×145 spatial dimension and 224 spectral bands. Its ground truth included 16 classes of vegetation. The Urban_F210 data set has images with 304×304 spatial dimension and 210 spectral bands. I choosed the ground truth with 2 classes of vegetation and 2 classes of buildings.

I choosed Urban_F210 because there were two classes owned by both of them (vegetation) and the other two not, it means they were not particularly similar but also not unrelated. Such characteristic is suitable for testing of transfer learning.

Our experiments runned on following hardware and software enviroments:

Table 3. Hardware and Software Environment (thanks to Mr.Gu's Computer)

Device Name	Configuration
GPU	Nvidia Titan RTX with 24GB of Memory
CPU	AMD Ryzen Threadripper 3960X 24-Core Processor
RAM	DDR4 3600MHz 128GB (32GB*4)
Language	Python
Platform	Tensorflow 1.3

4.2. Results

Before starting tests of Urban_F210 data set, I pretrained the original HybridSN model in IP data set for 6700 epoch. The learning rate was 0.01 and batch size was 256, after finishing the training, its overall accuracy (OA) was 99.84%, Kappa coeffcient (Kappa) was 98.49% and the average accuracy (AA) was 98.12%. I saved the parameters after training and loaded it when I tested fine-tuning methods in every model and their "frozen" rules given by TABLE II for 100 epoch. The setting of hyper parameters were the same.

Surpringly, only Network_5 and Network_7 showed similar accuracy performance as Network_original, all other fine-tuning methods showed poor performance and seemed to restrain the improvement of accuracy and the decrease of error.

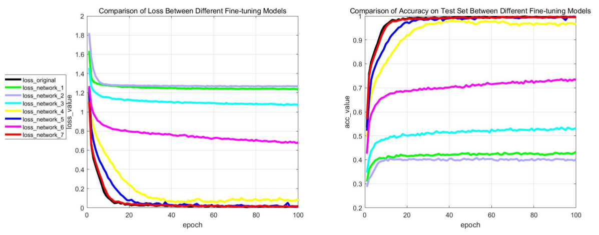

Figure 3. Comparision of Loss & Accuracy Between Different Fine-tuning Models.

Figure 3 showed the change of loss and accuracy of every model in training set within 100 epoch and TABLE IV was the classification accuracy in testing set; TABLE V shows the number of trainable parameters in different models:

4.3. Analysis

From the TABLE IV, it's obvious that none of those networks using fine-tuning can reach higher performance than starting training with none frozen layers, but there are still four models' accuracy going close to that of Network _Original : Network_7, Network_6, Network_5, Network _4.

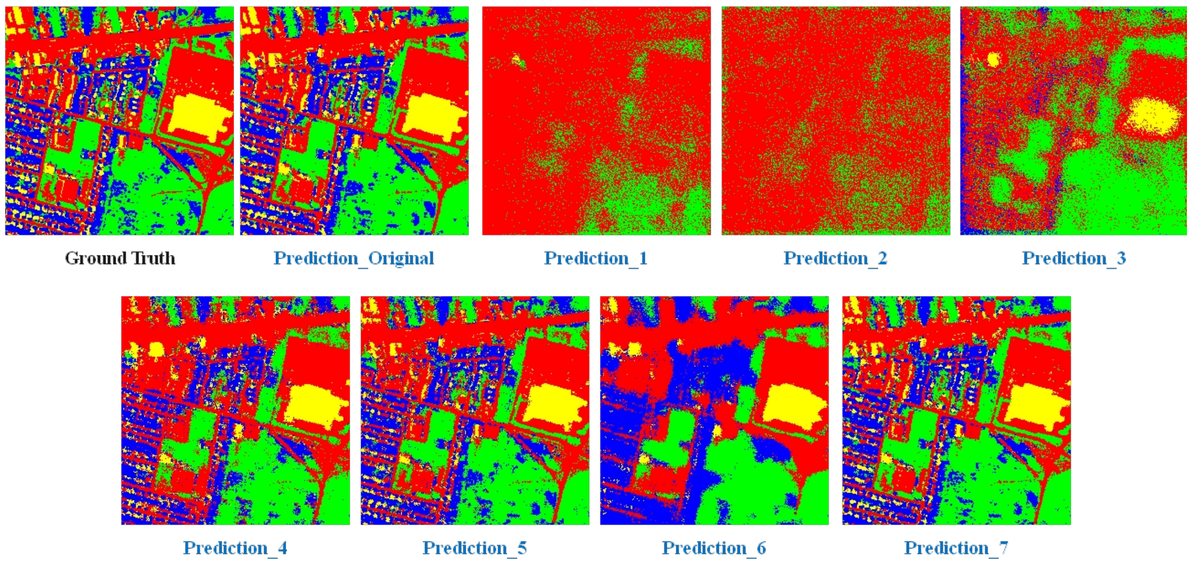

Figure 4. The prediction output pictures of different models

Table 4. Classification Accuracy in Testing Set After 100 Epoch

Models	Urban_F210		
	OA	Kappa	AA
Network_Original	83.68	76.84	81.35
Network_1	47.47	16.22	32.85
Network_2	44.14	10.87	30.17
Network_3	55.30	32.98	46.45
Network_4	75.03	63.99	70.23
Network_5	78.89	69.80	74.82
Network_6	72.11	60.34	67.46
Network_7	81.74	74.14	79.23

Table 5. Number of Trainable Parameters

Models	Trainable Numbers	Ratio
Network_Original	4,844,148.0	100%
Network_1	17,028.0	0.35%
Network_2	516.0	0.0011%
Network_3	33,412.0	0.69%
Network_4	4,768,644.0	98.44%
Network_5	4,824,004.0	99.58%
Network_6	55,876.0	1.15%
Network_7	4,837,860.0	99.87%

And from the TABLE V, we can easily find that Network_7 have the most trainable parameters and this is followed by Network_5 and Network_4, all of them consumed over 98% of parameters. Meanwhile, the ratio of Network_6 is only 1.15% but it performs similar accuracy as the above three after 100 epoch, according to TABLE IV.

An important conclusion can be summaried that a fine-tuning chooses to freeze fully-connected layer other than convolutional layer can help to achieve better performance.

Taking the two indicators into consideration, the fine-tuning method used on Network_6 which only train output layer and last convlutional layer seems to be the best choice whithin all these seven methods.

But why did none of those model perform better than original model ? I think the main reason is that our base model- HybridSN has already reached the level of the state of the art on HSI task of IP data set, it means that it fits such HSI task well and fine-tuning is even an disturbance to it.

And why didn't it perform as well as testing on IP data set or KCS data set? The reason comes from HSIs themselves, as I mentioned before, HSIs has data from hundreds of bands and in my model, it will pass a PCA layer before convolutional parts to filter out many waves; A lot of information about some specific bands will be dropped out. Important bands information of Urban_F210 may be useless and filtered in another data set. As I mentioned in the part of transfer learning, the divergence between target domain and source domain is the key to influence a transfer learning successful or not.

5. Conclusion

By comparing above indicators including test accuracy and number of parameters that needs training, I concluded that freeze dense layer may be a better choice instead of freezing convolutional layer while applying fine-tuning methods. Dramatical less computional units will be used through such method. Due to the complexity of HSIs, basic transfer learning like fine-tuning may perfrom terrible, advanced transfer learning methods like Deep Domain Confusion (DDC) [10] should be utilized. All in all, the combination of transfer learning and HSI is really an important and meaningful field. Just like the problems I mentioned in section 2, transfer learning has great potential to solve them.

6. Acknowledgement

This thesis is the final project of the course ”Introduction to Artificial Intelligence” by Professor Xichuan Zhou of Chongqing University. I feel thankful for his and teaching assistant’s guidance during this semester.

At the same time, I would also like to thank Mr. Jichen Gu for his Nvidia Titan RTX which helped me to implement my idea rapidly.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[2] Cheng Deng, Xianglong Liu, Chao Li, and Dacheng Tao. Active multi-kernel domain adaptation for hyperspectral image classification. *Pattern Recognition*, 77:306–315, 2018.

[3] Cheng Deng, Yumeng Xue, Xianglong Liu, Chao Li, and Dacheng Tao. Active transfer learning network: A unified deep joint spectral–spatial feature learning model for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(3):1741–1754, 2018.

[4] Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1086–1095, 2017.

[5] Thommen George Karimpanal and Roland Bouffanais. Self-organizing maps for storage and transfer of knowledge in reinforcement learning. *Adaptive Behavior*, 27(2):111–126, 2019.

[6] Shutao Li, Weiwei Song, Leyuan Fang, Yushi Chen, Pedram Ghamisi, and Jón Atli Benediktsson. Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6690–6709, 2019.

[7] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[8] Swalpa Kumar Roy, Gopal Krishna, Shiv Ram Dubey, and Bidyut B Chaudhuri. Hybridsn: Exploring 3-d–2-d cnn feature hierarchy for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 17(2):277–281, 2019.

[9] Viktor Slavkovikj, Steven Verstockt, Wesley De Neve, Sofie Van Hoecke, and Rik Van de Walle. Hyperspectral image classification with convolutional neural networks. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1159–1162, 2015.

[10] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.