Publications of the Astronomical Society of the Pacific, 132:044503 (10pp), 2020 April https://doi.org/10.1088/1538-3873/ab7548

Celestial Spectra Classification Network Based on Residual and Attention Mechanisms

Zhiqiang Zou[1,2], Tiancheng Zhu[1], Lingzhe Xu[3], and A-Li Luo[4,5]
[1] College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, 210023, People's Republic of China; zouzq@njupt.edu.cn
[2] Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing, Jiangsu, 210023, People's Republic of China
[3] Department of telescope's new technology, Nanjing Institute of Astronomical Optics & Technology, National Astronomical Observatories, CAS, Nanjing, Jiangsu, 210042, People's Republic of China
[4] Key Laboratory of Optical Astronomy, National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, People's Republic of China; lal@nao.cas.cn
[5] University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China
Received 2019 December 25; accepted 2020 February 11; published 2020 March 13

Abstract

In astronomy, it is important to categorize celestial bodies by classifying collected spectral data. The currently available methods present unsatisfactory spectral classification accuracy and incur high computing costs. We propose a celestial spectral classification network based on a residual and attention based convolutional network (RAC-Net). In this network, convolution operations can extract shallow and deep features of spectral data and classify them without relying on redshifts. The residual mechanism can augment the depth of the network and make training more efficient. The attention mechanism allows the network to focus on specific bands and specific features, rendering the learning more targeted. To evaluate the performance of the RAC-Net, we conducted a comparative test using a celestial spectral data set that consisted of 70,000 spectra collected by the large sky area multi-object fiber spectroscopic telescope. The experimental results showed that the classification accuracy of our network was up to 98.92%. Compared with the leading one-dimensional, convolutional neural network 1D SSCNN model, the RAC-Net presented higher classification accuracy and fewer network parameters.

Key words: miscellaneous – methods: data analysis – methods: statistical – stars: statistics – techniques: spectroscopic

Online material: color figures

1. Introduction

Human beings have been continuously exploring the universe. With the assistance of high-performance, astronomical telescopes, such as the Sloan digital sky survey (York 2000), the global astrometric interferometer for astrophysics (Perryman et al. 2001), and the large sky area multi-object fiber spectroscopic telescope (LAMOST) (Cui et al. 2012), it is possible to observe deeper parts of the universe. Although these telescopes can simultaneously observe thousands of celestial objects and collect spectra for research, the massive amounts of data cannot be effectively processed using traditional methods, because they normally present poor processing efficiency and accuracy. Fortunately, deep learning methods have provided us with solutions in the past decade; with deep learning, we can process these massive astronomical spectral data efficiently and accurately.

Currently available spectral classification methods can be roughly divided into two categories: pattern matching and machine learning methods (Corbally et al. 1994; Liu et al. 2015). The pattern matching method consists of finding highly recognizable spectral band features in the spectrum data and matching those features with typically known spectra by minimizing metric distances or maximizing similarities between them (Garrison 1984; LaSala 1994). The machine learning method consists of automatically learning all the features in the spectra and acquiring the relevant knowledge to then classify unknown spectra (Ball & Brunner 2010).

Presently, astronomers typically adopt the Morgan & Keenan (MK) classification method (Morgan & Keenan 1973), which divides stars into seven categories based on their temperature. A common step in processing spectra is to associate the spectrum with the MK class and then estimate the stellar astrophysical parameters. However, the traditional MK classification method requires manual or semi-manual comparison of spectra, which is inefficient and unreliable.

For high dimensional data, reducing the dimension while retaining useful information could decrease computing costs while keeping higher accuracy. Techniques, such as principal component analysis (PCA) and fisher matrix provide a linear method for reducing the dimension of the data. Jiang et al. (2013) discovered new cataclysmic variables by means of PCA

and the support vector machine (SVM) method. However, Liu et al. (2015) argued that MK automatic classification based on SVM is relatively poor in performance, and the way of directly classifying stars using line indices is possibly a more adequate choice.

The aforementioned methods, nonetheless, do not capture the nonlinear characteristics of the celestial spectra accurately; therefore, nonlinear methods, such as nonlinear PCA, information bottleneck (Slonim et al. 2001), and artificial neural networks (ANNs) have been proposed. Since the publication of von Hippel et al. (1994) and Singh et al. (1998), ANNs began to play an important role in astronomy. Vieira & Ponz (1998) used ANNs and self-organizing maps to obtain good correlations compared with the ground-truth from manual classification. In Bailer-Jones (1997), stellar parameters (such as effective temperature and surface gravity) were successfully obtained from the spectrum by ANNs. Wang et al. (2016) proposed a fast, layer-wise learning algorithm based on ANNs that increased classification accuracy to 0.8232.

In 2006, Hinton introduced the concept of deep learning (Hinton et al. 2006), which has made positive contributions in various fields. Fabbro et al. (2018) constructed deep neural networks and obtained a root mean square error of 0.04 when predicting stellar parameters. Pasquet-Itam & Pasquet (2018) achieved a precision of 0.988 in the detection of quasars by means of a deep learning approach. Hon et al. (2017) used a one-dimensional, convolutional neural network to classify red giant stars with an accuracy of 99%. Liu et al. (2019) established a nine-layer, convolutional network and attained an accuracy of 90%, 93%, and 97% in the classification of F, G, and K stars, respectively.

However, there are still problems to overcome in the available deep learning methods for classifying spectra; first, most of the models are not deep enough, which leads to insufficient extraction of features. In general, the deeper the model, the more in-depth features can be extracted; nonetheless, deeper models have drawbacks, such as gradient dispersion or explosion that could increase the difficulty of the training model. Second, the large number of parameters needed result in high computing costs. Third, the training of the model lacks pertinence, i.e., there is not enough attention to important spectral bands during the training.

In this study we investigate a residual and attention based convolutional network (RAC-Net), in which residual (He et al. 2016) and attention mechanisms (Chen et al. 2017; Vaswani et al. 2017; Hu et al. 2018; Woo et al. 2018) are introduced. The residual mechanism solves the problem of vanishing or exploding gradients by employing shortcut connections between the inputs and outputs of the residual block; this could also reduce the parameter quantity of the model. In addition, the attention mechanism improves the pertinence of the model during training; it can automatically pay the attention to the features that benefit classification results while ignoring invalid features, so that the interpretability of the model is enhanced.

This paper is organized into the following sections: In Section 2, we will explain our data and methods. The experimental results and comparisons with other models are presented in Section 3. Finally, Section 4 is our conclusion, where we will discuss the advantages and disadvantages of RAC-Net.

2. Materials and Methods

2.1. Data Set Description

We used spectral data from LAMOST, which is publicly available on LAMOST's official data website.[6] Each spectrum consists of approximately 3800 dimensional data with a wavelength ranging from 3690 to 9100 Å. In the training data set, each spectrum had a corresponding label including a main class (STAR, GALAXY, QSO, or UNKNOWN) and a subclass (MK class). The spectral resolution was approximately 1800 (Cui et al. 2012). The elements in the UNKNOWN class refer to bad quality spectra with low confidence in template matching, and are, therefore, not classified under the other three categories. In this study, we used 70,000 pieces of spectral data to construct three data sets. The training and test sets in each data set had a division ratio of 8:2 and were randomly divided.

2.2. Data Preprocessing

In machine learning methods, data pre-processing is very important. Our data pre-processing method was divided into the following three steps: (1) One-hot encoding for labels. For the four celestial objects (STAR, GALAXY, QSO, and UNKNOWN), by using one-hot encoding method in machine learning methods (Hu et al. 2018), we encode them into "0001," "0010," "0100," and "0100," respectively. The same operation was performed for the MK class labels. (2) Flux Standardization. Flux standardization was used to normalize spectral values between 0 and 1 based on the flux intensity of each spectrum. This operation ensures that the attenuation of celestial light during propagation does not affect the learning of the model (Li et al. 2007). The specific formula for flux standardization is shown in Equation (1):

$$x_i = \frac{x_i}{\|x\|_2} \tag{1}$$

where x_i represents the ith spectral sample in the data set, and $\|x\|_2$ represents the two-norm of the sample, i.e., the flux of the spectrum.

(3) Data Augmentation. The collected spectral data presented an uneven distribution; 90% of the objects were included in the STAR class, whereas the QSO class was the class with the

[6] http://dr5.lamost.org/

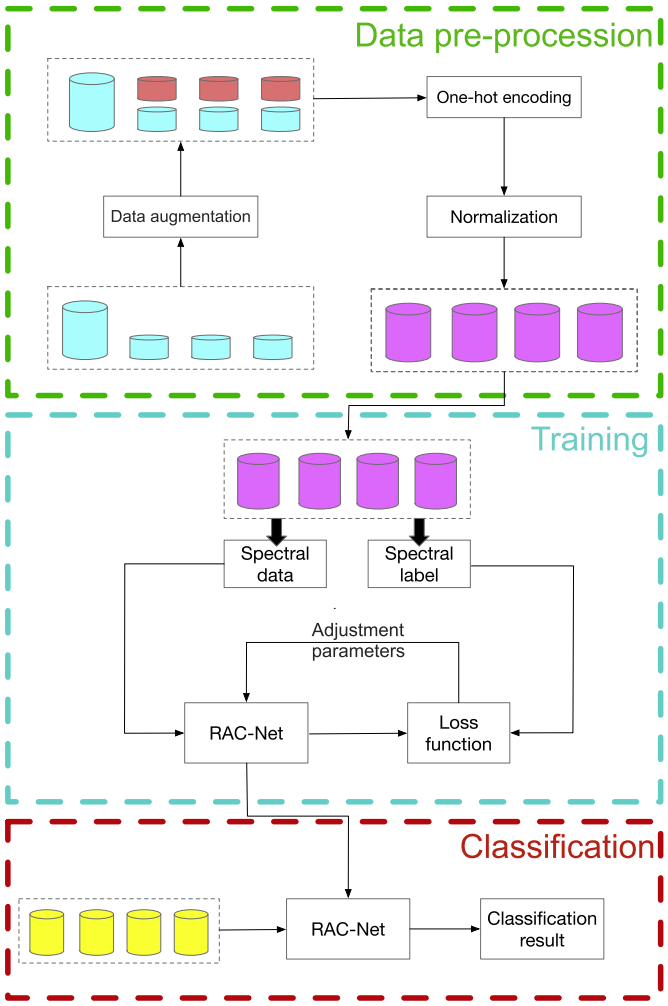

Figure 1. Framework of the RAC-Net.
(A color version of this figure is available in the online journal.)

fewest elements (1%). This would have been unfavorable for classification, as the learning of the model would have focused on the STAR class data while ignoring the other classes. To tackle this problem, we performed data augmentation by down-sampling the STAR class data and up-sampling the other three classes. Further, we added random Gaussian noise into the resampled data to enhance the variability of training data. We used a similar approach to resample the data set for MK class classification in order to keep the data balanced.

2.3. Framework of the RAC-Net

The framework of the RAC-Net is shown in Figure 1. This network consisted of three essential components: data pre-processing, training, and classification. First, the data pre-processing component performed data enhancement, encoding, and standardization operations, as discussed in Section 2.2. Afterwards, the focus was put on the second component (the training component), in which the required knowledge for classification was learned from the labels by minimizing the loss of the real label and the predicted result. After this step, our supervised learning-based model could classify unlabeled spectra with high accuracy and efficiency without any prior knowledge on astronomy. Finally, the classification component classified the unlabeled spectral data by using the completed RAC-Net. There were three core parts in the RAC-Net: the convolutional network, the residual block, and the attention block. These three parts are addressed in Sections 2.4–2.6.

2.4. Convolutional Network

Artificial neural networks can be divided into input, hidden, and output layers. Among them, the more hidden layers, the deeper the model and the more valuable the features that can be extracted. As these hidden layers continue to stack, the nonlinear activation functions in the neurons are also constantly nested to form more complex nonlinear functions. Training is the continuous process of fitting these nonlinear functions to the data-label mapping function.

In the hidden layers, we used convolution to extract local information of the data and therefore, identify local features of the spectra. This operation was performed by sliding the convolution filter over the data and computing the inner product with it; the more similar the spectral data was to that of the filter, the stronger the resulting response was. Convolution filters with different shapes were used on each hidden layer to extract different local features. The convolution operation of our RAC-Net is shown in Equation (2):

$$y_{i\in(0,L-l+1)} = f\left(\sum_{j=0}^{l} x_{i+j} * c_j + b\right) \tag{2}$$

where L and l represent the length of the spectrum data x and the convolution filter parameter c and b represents the offset after the convolution operation, and $f(.)$ represents the nonlinear activation function. We used the *ReLU* function, which is shown in Equation (3). The variable y represents the result of the data after a convolution operation. An example of a specific convolution operation is shown in Figure 2.

$$ReLU(x) = \max(x, 0). \tag{3}$$

As can be seen in Figure 2, a local feature of a spectrum can be converted into a sequence, such as [3, 2, −1, 2, 0, 0, −1, 2], and then a filter, such as [−1, 2, 0], can be used to slide over it. By observing the resulting sequence, [1, −4, 5, −2, 0, −2], it can be seen that, when the filter slides to the third element of the sequence, the response reaches a maximum, indicating that the local feature and the position have been extracted. This local feature extracting method can be also used to reduce the influence of redshift on the spectrum because the redshift effect consists of a distance movement in the infrared direction, while the convolution operation sliding over the spectral data depends

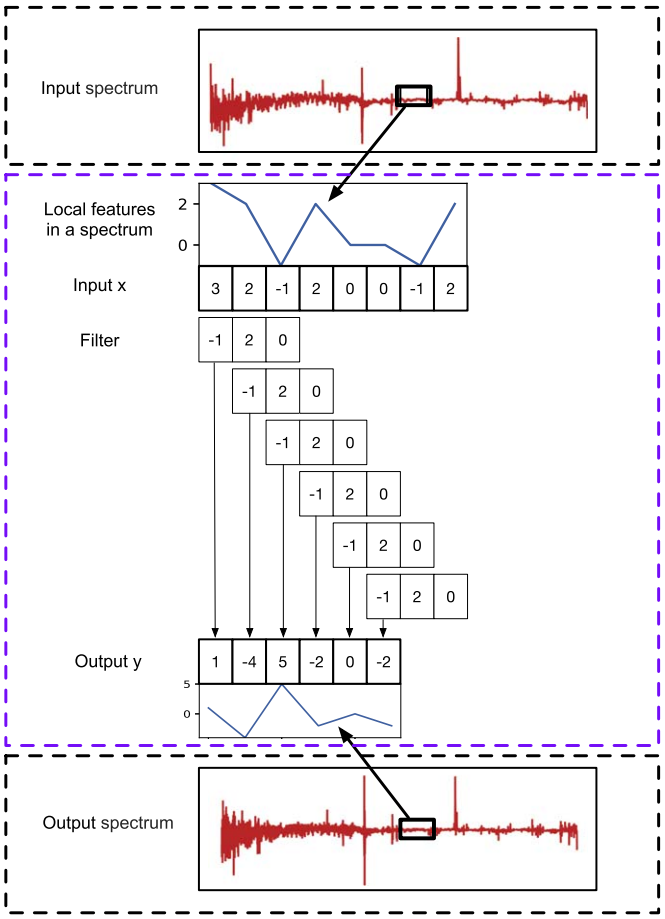

Figure 2. Representation of the convolution operation applied to input spectra. (A color version of this figure is available in the online journal.)

on the relative relationship between neighbor elements, and not on absolute relationships. Therefore, redshift has less influence on feature extraction when convolution filters are used.

The convolutional layer output, i.e., the convolutional feature, is processed by the fully connected (FC) layer. This layer performs a global analysis on the convolutional feature, and its output corresponds to the classification result of the spectrum. The operation of the FC layer is represented in Equation (4):

$$Y_{i,n} = \mathrm{softmax}(W * y_i + b)_n \tag{4}$$

where $Y_{i,n}$ represents the probability that the ith datum belongs to category n, W is the parameter of the FC layer network, y_i is the convolutional feature of the ith datum, and b is an offset parameter. The expression of the $\mathrm{softmax}(.)_n$ function is presented in Equation (5):

$$\mathrm{softmax}(x)_n = \frac{e^{x_n}}{\sum_{k=0}^{K} e^{x_k}} \tag{5}$$

where K is the length of the spectrum data x.

In the training process, we used the backpropagation (BP) algorithm to adjust the parameters of the model. The steps of the BP algorithm were as follows: first, the training data was forwarded to all the layers, and the loss was calculated by comparing the predicted label with the true label. Then, the loss was propagated backward to all the layers, and the parameters of each layer were adjusted by the gradient descent method. The loss function used as the cross-entropy loss function, as shown in Equation (6):

$$\mathrm{loss}_i = -\sum_{n=1}^{4} Y_{i,n} \log(Y_{i,n}^{\mathrm{true}}) \tag{6}$$

where $Y_{i,n}^{\mathrm{true}}$ represents the nth value of true label of the ith datum. Since our labeling system was based on one-hot encoding, each true label consisted of a four-digit number in which the corresponding value was "1" and the others values were "0."

2.5. Residual Block

In deep learning, the deeper the model, the closer the outcome is to the true label. However, in actual operations, with the depth augmentation, the vanishing gradient and the exploding gradient would appear, which are the main factors that hinder the model learning. Vanishing gradient and exploding gradient refer to situations in which the gradient information used for transmission is too small or too large in the process of network parameter adjustment, leading to the failure of network training.

In the training process of the model, the BP algorithm was used to propagate the loss backward. The parameters were adjusted by following the chain rule, in which the gradient of the nonlinear activation function has to be continuously propagated. As the activation function of each layer was nonlinear rather than linear function, the multiplication of the gradient led to vanishing or exploding phenomena in the propagation process when the number of network layers increased. Therefore, to solve the problem of gradient dispersion and gradient explosion in the deep learning model, He et al. (2016) proposed the use of a residual mechanism performed by the residual block. A residual block is composed of several convolutional layers; however, in contrast to the traditional structure, a shortcut connection is added between the input and output of the block. The structure of residual block in our RAC-Net is shown in Figure 3.

In Figure 3, input_x is directly transmitted to the output by the shortcut connection. The output is $H(\mathrm{input\_}x) = F(\mathrm{input\_}x) + \mathrm{input\_}x$. When $F(\mathrm{input\_}x) = 0$, then $H(\mathrm{input\_}x) = \mathrm{input\_}x$, which is the identity map. Therefore, the residual block changes the learning target from a complete output to the so-called residual: $F(\mathrm{input\_}x) = H(\mathrm{input\_}x) - \mathrm{input\_}x$. This structure presents three benefits:

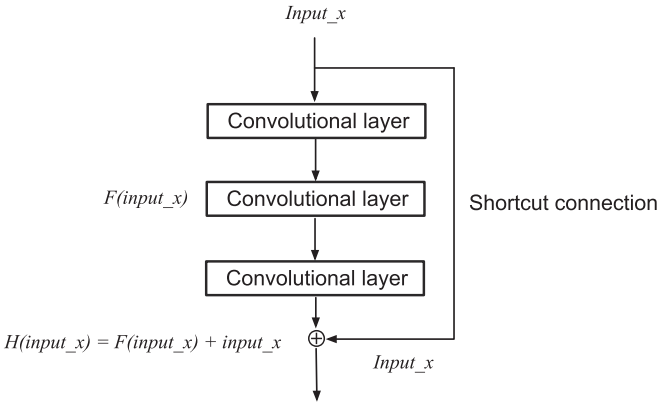

Figure 3. Residual block structure in RAC-Net.

(1) Instead of being multiplied by the activation function, the gradients propagated backward from the deeper layer can be passed directly to the shallower layer via the shortcut connections so that the gradient does not vanish or explode.
(2) When the output of a certain layer has reached the optimal result, the subsequent layers can be directly trained as, so that the subsequent layers become the identity map. Therefore, the complexity of the model can be adjusted to be optimal.
(3) From the perspective of celestial spectral data, the depth of the model is greatly augmented by using the residual block. In consequence, the fitted function becomes more consistent with the characteristics of celestial spectral data.

2.6. Attention Block

The attention block allows our RAC-Net to focus on certain features. When we are observing a particular situation (i.e., a picture), the degree of attention we put on the "target" (i.e., the runner in the picture) is different from that we put on the background; our attention would be always focused on the runner, which is often the most important local feature of the scene. Similarly, in spectral data, different bands carry different information. In the process of light propagation, spectral information is greatly affected by the background of the universe and the Earth's atmosphere, particularly in certain specific bands. If we could find the bands that are severely affected, we would pay less attention to these bands. That is, we would pay more attention to important bands and improve the performance of the model.

In convolution operations, each convolution filter produces a channel. Traditional convolutional networks treat different channels as if they contributed equally to the subsequent classification results. However, as what presented in Hu et al. (2018), Woo et al. (2018), different channels carry different characteristics: while some of these features contribute greatly to classification, others only provide small contributions. Moreover, some features can have a negative impact. Therefore, we introduced an attention block to assign weight to different channels and thus allocate a higher weight to important channels. The structure of channel attention is shown in the left half of Figure 4.

The attention mechanism was implemented by employing attention blocks. In an attention block, the state of the data can be divided into eight stages as shown in the right half of Figure 4. In Stage 1, a convolution operation is performed on the input spectrum, resulting in c channels (Stage 2). The *MaxPooling* and *AveragePooling* are respectively the operation of taking the maximum value and the average value on the features after convolution.Then the *MaxPooling* and *AveragePooling* functions operate on these c channels, resulting in two vectors with lengths equal to c (Stage 3), as described in Equations (7) and (8)

$$w_c^{\max} = \max_l(y_{l,c}) \tag{7}$$

$$w_c^{\text{average}} = \frac{1}{l}\sum_{i=0}^{l} y_{l,c} \tag{8}$$

where $y_{l,c}$ represents the data with length l and channel c in Stage 2. The variables $w_c^{\max}$ and w_c^{average} are vectors generated after *MaxPooling* and *AveragePooling*, respectively, and correspond to the two cuboids produced in Stage 3.

Further, the two vectors from Stage 3 are concatenated to one vector w_c in Stage 4, as presented in Equation (9):

$$w_c = [w_c^{\max}, w_c^{\text{average}}]. \tag{9}$$

Subsequently, the initial weight W_c is sent to an FC network. After learning and adjustment, the output is the final weight w_c^{att}, which is the attention weight (Stage 5) described in Equation (10):

$$w_c^{\text{att}} = f(W * w_c + b). \tag{10}$$

At the *RepeatVector* operation, the attention weight is extended to an attention matrix with $l * c$ dimension in Stage 6.

In Stage 7, the attention matrix is multiplied by the convolution data from Stage 2 according to Equation (11), and the product y_{out} is the output of the attention block (Stage 8).

$$y_{\text{out}} = w_c^{\text{att}} * y_{l,c}. \tag{11}$$

2.7. Experimental Setting

We used three data sets in our experiment. Data set 1 was made up of 40,000 spectra of STAR, GALAXY, QSO, and UNKNOWN, in which the number elements for each class was 10,000. Data set 2 was based on Data set 1, but without 10,000 spectra whose class was UNKNOWN (Data set 2 had 30,000 spectra equally distributed among three classes). Data set 3 was

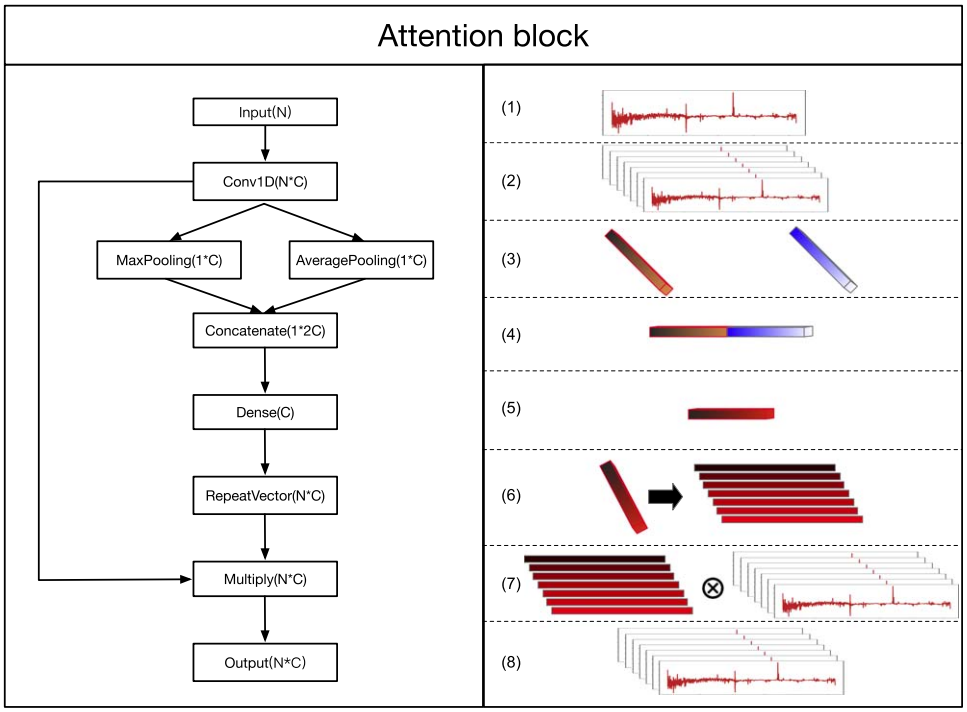

Figure 4. Attention block structure in RAC-Net.
(A color version of this figure is available in the online journal.)

Table 1
Experimental Data Sets

Data Sets	Class 1	Class 2	Class 3	Class 4
Data set 1	STAR:10 000	GALAXY:10 000	QSO:10 000	UNKNOWN:10 000
Data set 2	STAR:10 000	GALAXY:10 000	QSO:10 000	UNKNOWN:0
Data set 3	F:10 000	G:10 000	K:10 000	···

Note. Data taken from the LAMOST.

the data for MK classification; it consisted of three classes (F, G and K) with 10,000 spectra each. The composition of each data set is shown in Table 1.

The construction procedure of the RAC-Net is presented gradually in Figures 5–7. There were two types of blocks in our RAC-Net: residual blocks and attention blocks. Residual blocks consisted of three convolution layers and one attention block. The hidden layer of the model consisted of eight residual blocks. The number of convolution filters per convolutional layer incremented layer by layer to extract as many deep features as possible. After the convolutional layers, the convolution features were sent to an FC layer of 128 neurons, in which the neurons were randomly put in a dormant state (such as the gray neurons in Figure 5) with a probability of 0.5 to prevent over-fitting (Labach et al. 2019).

The model for the RAC-Net was created using Python 3.5 as the programming language and the Keras deep learning framework. The CPU used was an Intel® Core™ i7-4790 CPU @ 3.60 GHz, the RAM memory was 8 GB, and the hard disk was a 256 GB solid-state drive.

3. Results and Discussion

3.1. Performance of the Convolutional Network (C-Net)

We first built a plain network, C-Net, that consisted of eight convolutional layers and max-pooling layers. The structure of the C-Net is shown in Figure 5. The number and size of convolution filters were consistent with our RAC-Net. Data set 1 was used to compare the classification accuracy of the C-Net with those of the normal FC network in Zou et al. (2019).

As it can be observed in Table 2, the classification accuracy of the C-Net was higher than that of the FC network, particularly for the UNKNOWN class. This was observed for the classification accuracy for four classes and their average

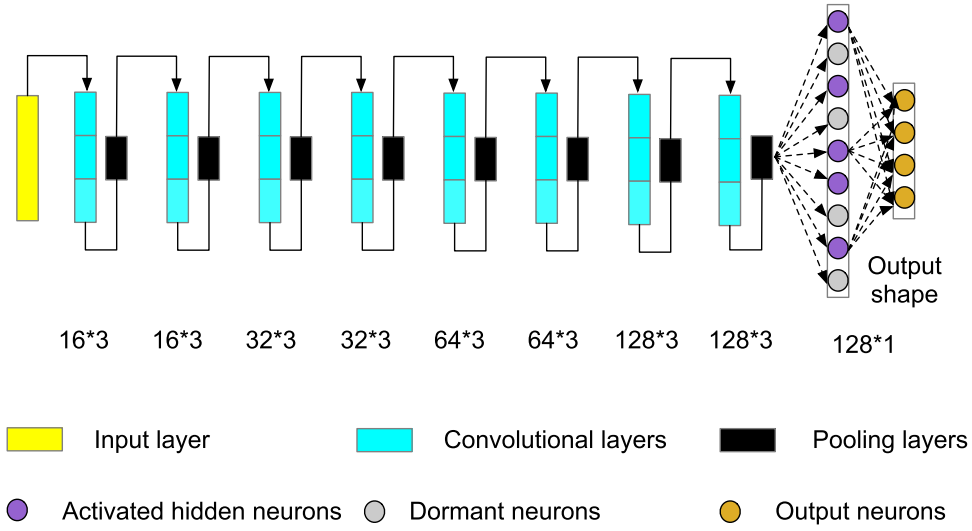

Figure 5. Structure of the C-Net.
(A color version of this figure is available in the online journal.)

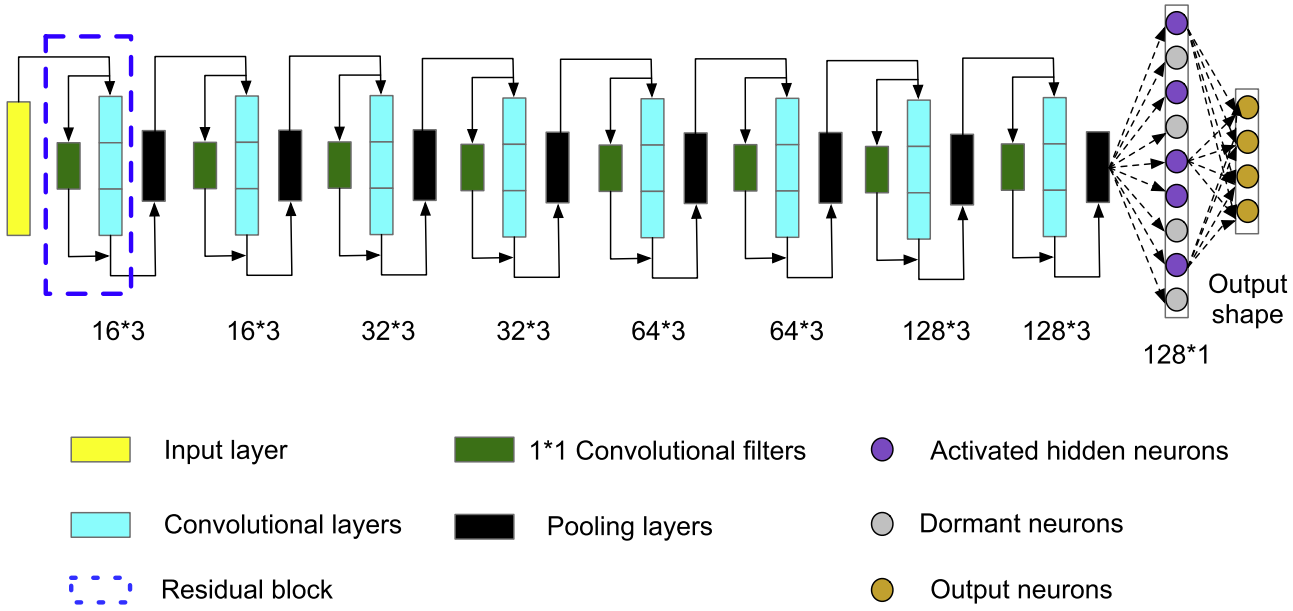

Figure 6. Structure of the residual based convolutional network (RC-Net).
(A color version of this figure is available in the online journal.)

accuracy. Compared with the C-Net, it was difficult for the FC network to identify the characteristics of the UNKNOWN class. This shows that the convolutional network was more capable of identifying spectral features than the FC network.

3.2. Performance of the Residual Block (RC-Net)

Based on the C-Net of Section 3.1, we replaced three convolutional layers with a residual block to optimize the deep convolutional network and created the RC-Net (Figure 6). To compare our model with the one-dimensional, convolutional neural network (1D SSCNN) model (Liu et al. 2019), we separately conducted comparison experiments on three data sets. The experimental results are shown in Table 3.

Overall, the three models performed better on Data set 2. There are two reasons behind this phenomenon: first, by comparing the results for Data set 2 with those for Data set 1, it can be seen that the performance of the models was improved

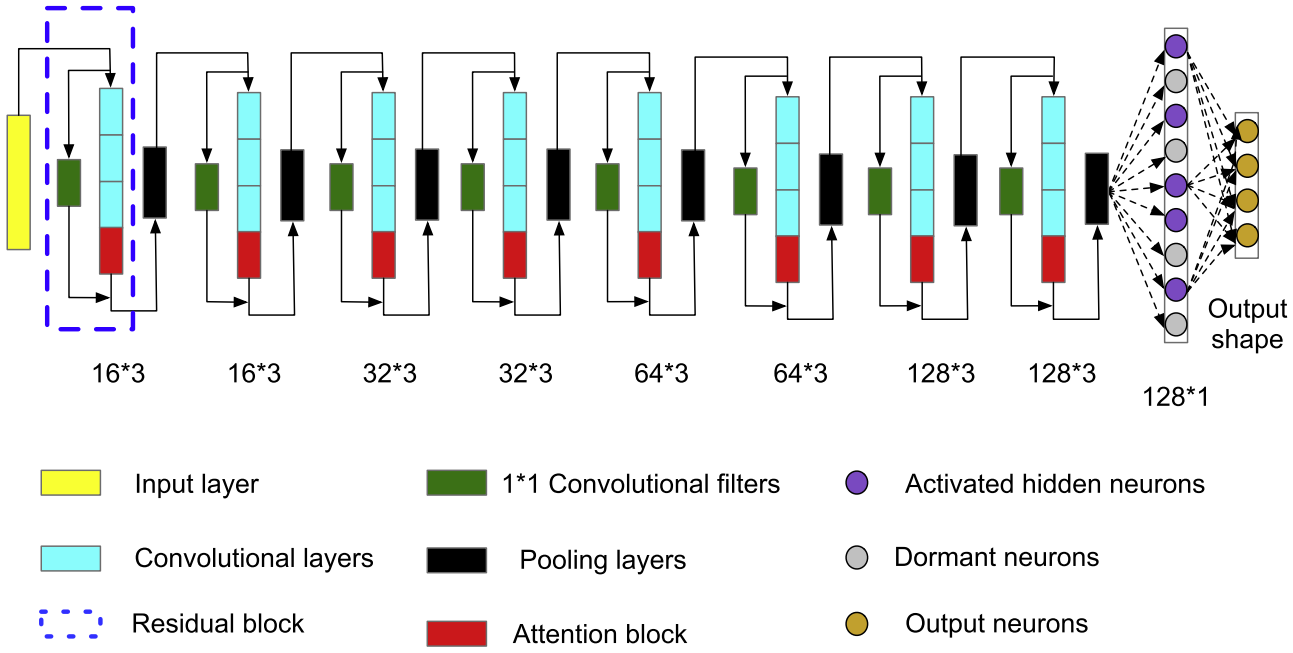

Figure 7. Structure of the residual based convolutional network (RC-Net).
(A color version of this figure is available in the online journal.)

Table 2
Classification Accuracy of the FC Network and the C-Net for Different Classes

Model	STAR	GALAXY	QSO	UNKNOWN	Average Accuracy
FC network	91.38%	83.43%	85.65%	48.17%	77.13%
C-Net	**91.50%**	**94.41%**	**90.16%**	**79.91%**	**88.75%**

Note. The bold value in table indicates the model with the best performance.

Table 3
Classification Accuracy of the C-Net, the 1D SSCNN, and the RC-Net

Model	Data Set 1	Data Set 2	Data Set 3
C-Net	88.75%	98.58%	91.97%
1D SSCNN	92.85%	98.45%	92.33%
RC-Net	**93.33%**	**98.81%**	**93.07%**

Table 4
Classification Accuracy of the C-Net, the 1D SSCNN, and the RC-Net

Model	Parameters	Data Set 1	Data Set 2	Data Set 3
1D SSCNN	5.18M	92.85%	98.45%	92.33%
C-Net	2.89M	88.75%	98.58%	91.97%
RC-Net	**0.388M**	93.33%	98.81%	93.07%
RAC-Net	0.623M	**93.52%**	**98.92%**	**93.25%**

significantly after removing the interference class UNKNOWN. As the quality of the UNKNOWN class data contained in Data set 1 was poor, this data interfered greatly with the model's learning for the other three categories. Second, by comparing the results for Data set 2 with those for Data set 3, it can be seen that the classification for celestial classes is easier than that for MK classes. In addition, our RC-Net performed better than the C-Net with the same depth. This was because the residual block allowed the model to be better trained.

3.3. Performance of the Attention Block (RAC-Net)

We constructed our RAC-Net as shown in Figure 7 and compared its performance with that of other models. The experimental results on accuracy are shown in Table 4.

As can be seen in Table 4, after we inserted the attention blocks, the accuracy was improved by approximately 0.2% compared with the results presented in Section 3.2. This was similar to the improvement effect promoted by the attention mechanism reported in other papers (Hu et al. 2018; Woo et al.

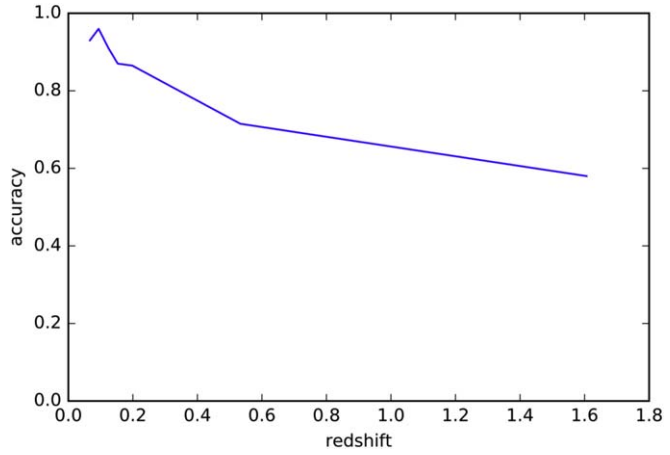

Figure 8. Effect of redshift on classification accuracy for Data set 3. (A color version of this figure is available in the online journal.)

2018). The reason why the improvement caused by the attention mechanism was not high might have been that the contribution of different channels to classification accuracy was close.

In addition, the number of parameters in our model was approximately 623,000, which is only 12% of that of the 1D SSCNN model. Therefore, the experiments empirically show that the RAC-Net is an effective model to improve the performance of the 1D SSCNN model using fewer parameters.

3.4. Performance Under Different Redshift Conditions

The characteristics of spectra change under the influence of the redshift effect. As for the pattern matching method (Garrison 1984; LaSala 1994), redshift would affect the classification results. In comparison with these methods, redshift conditions have less influence on our RAC-Net, which can complete the classification without knowing the redshift effect. Figure 8 shows the classification accuracy for Data set 3 under different redshift values.

It can be observed from Figure 8 that when we use high-redshift data from GALAXY or QSO classes, the classification accuracy decreases significantly. This is because the spectrum is stretched when the redshift effect is intense enough, leading to the complete change of the wavelength meaning in the whole observation band.

4. Conclusions

In this paper, we proposed a deep convolutional network (RAC-Net) by introducing residual and attention blocks to a plain network. Owing to the properties of the convolution operation, the RAC-Net was capable of classifying spectra without considering redshift. Residual blocks overcame the difficulties related to vanishing gradients and exploding gradients and extracted more deep features than plain networks. Attention blocks increased the focus of the RAC-Net on the features that contribute more important information, and thus made the RAC-Net pay more attention to important channels. In addition, the total computing cost is expected to decrease owing to the reduced number of parameters of the RAC-Net.

In conclusion, the experiments showed that the RAC-Net has some advantages over other models on different data sets. However, our model still presents limitations when dealing with data containing UNKNOWN type elements. This is because the model is sensitive to noise. Therefore, the noise is misinterpreted by the RAC-Net as a feature to be learned, which would disturb the training on normal features. Moreover, the classification of GALAXY or QSO classes with the presence of high redshift is still a challenge because the wavelength meaning of input spectra is distorted. Subsequent work will focus on finding a superior model to identify outlier spectra and classify high-redshift data.

This work is supported by the National Natural Science Foundation of P. R. China (No. 41601449), the Key Laboratory of Spatial Data Mining & Information Sharing of Ministry of Education, Fuzhou University (No. 2016LSDMIS07), and the Joint Research Fund in Astronomy (grant No. U1931209, grant No. U1931207) under cooperative agreement between the National Natural Science Foundation of China and Chinese Academy of Sciences. Guoshoujing Telescope (the Large Sky Area Multi-Object Fiber Spectroscopic Telescope LAMOST) is a National Major Scientific Project built by the Chinese Academy of Sciences. Funding for the project has been provided by the National Development and Reform Commission. LAMOST is operated and managed by the National Astronomical Observatories, Chinese Academy of Sciences. In addition, we are grateful to the anonymous reviewers for their insightful and constructive suggestions.

Software: Numpy (Oliphant 2006), Matplotlib (Hunter 2007), Keras (Chollet et al. 2018), Tensorflow (Abadi et al. 2015).

ORCID iDs

Zhiqiang Zou, https://orcid.org/0000-0003-2828-8491
A-Li Luo, https://orcid.org/0000-0001-7865-2648

References

Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-scale Machine Learning on Heterogeneous Systems, http://tensorflow.org/
Bailer-Jones, C. A. L. 1997, PASP, 109, 932
Ball, N. M., & Brunner, R. J. 2010, IJMPD, 19, 1049
Chen, L., Zhang, H., Xiao, J., et al. 2017, in Proc. IEEE Conf. Computer Vision and Pattern Recognition *(Honolulu, HI, 21–26 July 2017)* (Piscataway, NJ: IEEE), 5659
Chollet, F. 2018, Keras: The Python Deep Learning library, Astrophysics Source Code Library, ascl:1806.022

Corbally, C. J., Gray, R. O., & Garrison, R. F. 1994, in Proc. of a Workshop of the Vatican Observatory, vol 60 *(Tucson, AZ, September 1993)* (San Francisco, CA: ASP)
Cui, X.-Q., Zhao, Y.-H., Chu, Y.-Q., et al. 2012, RAA, 12, 1197
Fabbro, S., Venn, K. A., O'Briain, T., et al. 2018, MNRAS, 475, 2978
Garrison, R. F. 1984, The MK Process and Stellar Classification
He, K., Zhang, X., Ren, S., & Sun, J. 2016, in Proc. IEEE Conf. Computer Vision and Pattern Recognition *(Las Vegas, NV, 27-30 June 2016)* (Piscataway, NJ: IEEE), 770
Hinton, G. E., Osindero, S., & Teh, Y.-W. 2006, Neural Comput., 18, 1527
Hon, M., Stello, D., & Yu, J. 2017, MNRAS, 469, 4578
Hu, J., Shen, L., & Sun, G. 2018, in Proc. IEEE Conf. Computer Vision and Pattern Recognition *(Salt Lake City, UT, 18–23 June 2018)* (Piscataway, NJ: IEEE), 7132
Hunter, J. D. 2007, CSE, 9, 90
Jiang, B., Luo, A., Zhao, Y., & Wei, P. 2013, MNRAS, 430, 986
Labach, A., Salehinejad, H., & Valaee, S. 2019, arXiv:1904.13310
LaSala, J. 1994, in Proc. of a Workshop of the Vatican Observatory *(Tucson, AZ, September 1993)* ed. C. Corbally, R. O. Gray, & R. F. Garrison, 312
Li, X., Liu, Z., Hu, Z., Wu, F., & Zhao, Y. 2007, Guang Pu Xue Yu Guang Pu Fen Xi = Guang Pu, 27, 1448
Liu, C., Cui, W.-Y., Zhang, B., et al. 2015, RAA, 15, 1137
Liu, W., Zhu, M., Dai, C., et al. 2019, MNRAS, 483, 4774
Morgan, W. W., & Keenan, P. C. 1973, ARA&A, 11, 29
Oliphant, T. E. 2006, A Guide to NumPy, Vol. 1 (USA: Trelgol Publishing)
Pasquet-Itam, J., & Pasquet, J. 2018, A&A, 611, A97
Perryman, M. A. C., de Boer, K. S., Gilmore, G., et al. 2001, A&A, 369, 339
Singh, H. P., Gulati, R. K., & Gupta, R. 1998, MNRAS, 295, 312
Slonim, N., Somerville, R., Tishby, N., & Lahav, O. 2001, MNRAS, 323, 270
Vaswani, A., Shazeer, N., Parmar, N., et al. 2017, in Advances in Neural Information Processing Systems *(Long Beach, CA, 4–9 December 2017)* (San Diego, CA: NeurIPS), 5998
Vieira, E. F., & Ponz, J. D. 1998, in ASP Conf. Ser. 145, Astronomical Data Analysis Software and Systems VII, ed. R. Albrecht, R. N. Hook, & H. A. Bushouse (San Francisco, CA: ASP), 508
von Hippel, T., Storrie-Lombardi, L. J., Storrie-Lombardi, M. C., & Irwin, M. J. 1994, MNRAS, 269, 97
Wang, K., Guo, P., & Luo, A.-L. 2016, MNRAS, 465, 4311
Woo, S., Park, J., Lee, J.-Y., & So Kweon, I. 2018, in Proc. European Conf. Computer Vision (ECCV) (Berlin: Springer), 3
York, D. G., Adelman, J., Anderson, J. E., et al. 2000, AJ, 120, 1579
Zou, Z., Zhu, T., & Xu, L. 2019, in 2019 4th Int. Conf. Computational Intelligence and Applications (ICCIA), IEEE *(Nanchang, 21–23 June 2019)* (Piscataway, NJ: IEEE), 68