Identification of Interference Sources in Spiral Galaxy Images Based on Convolutional Neural Network

Shiqing Wu\textsuperscript{1,a} Tao Hu\textsuperscript{2,b,*}

\textsuperscript{1}Shanghai University, No. 99 Shangda Road, Baoshan District, Shanghai, China
\textsuperscript{2}Shanghai University, No. 99 Shangda Road, Baoshan District, Shanghai, China
\textsuperscript{a}Shi0366742@qq.com \textsuperscript{b}hysjzxy@163.com

*Corresponding author.

ABSTRACT: In the spiral galaxy image, the luminosity of the precursor star due to its close proximity to the spiral galaxy reaches an approximate order of magnitude, which makes the precursor stars become an important interference source in the galaxy image and makes the computer difficult to distinguish them. Therefore, it is of great significance to remove the precursor stars from the galaxy image. In this paper, a modified convolution neural network (CNN) is used to classify spiral galaxies and stars in order to find pre-star interference. Convolutional neural network (CNN) is a popular image classification technology, which is included in depth learning. It solves the problem of a large amount of data and serious non-linearity on the basis of neural network. Based on CNN, this paper studies and improves the selection of network level, the optimization algorithm, batch normalization and so on. The two types of astronomical images of the target are trained and predicted, and the classification results are very good. At present, the error rate is around 3%.

1. INTRODUCTION

Neural network simulates the process of neuron information transmission, and optimizes the parameters needed in the process of transmission by back propagation of errors. However, image carriers with huge data make it difficult for neural networks to train and optimize the parameters of such problems, and may lead to more problems such as over-fitting. Three years after the BP (Back Propagation) algorithm\textsuperscript{1} was proposed, LeCun chose to use BP algorithm to train multi-layer convolution neural network (CNN) to recognize handwritten numerals. This can be said to be the embryonic form of CNN, but the effect is unsatisfactory. This is the earliest literature proposed by the concept of convolution neural network (CNN), but it has not been specifically implemented\textsuperscript{2}. In 1998, the LeNet-5 model\textsuperscript{3}, still proposed by LeCun, marked the formal formation of CNN. CNN uses convolution nucleus, which is similar to receptive field to make local information high-level and then extract information. Compared with traditional neural network, CNN greatly reduces the number of neurons in the input layer and speeds up the operation of the whole network.

Because of its close distance, the pre-star produces an approximate order of magnitude of luminosity with spiral galaxies, which makes the pre-star an important interference source in galaxy images and makes it difficult for computers to distinguish it. In this paper, the image of spiral galaxies and stars is classified based on improved convolution neural network, and the interference source of pre-star in spiral galaxy images is eliminated effectively.
2. RELEVANT ALGORITHMS OF CONVOLUTIONAL NEURAL NETWORKS

2.1 Convolutional Neural Network\cite{4}

CNN is a feedforward neural network. Its artificial neurons can respond to a part of the surrounding units in the coverage area and perform well in large-scale image processing. Generally, the basic structure of CNN consists of two layers. One is feature extraction layer. The input of each neuron is connected with the local acceptance domain of the previous layer, and the local features are extracted. Once the local features are extracted, the position relationship between the local features and other features is determined. The other is feature mapping layer, network. Each computing layer consists of multiple feature maps, each feature map is a plane, and the weights of all neurons on the plane are equal. Because the neurons on one mapping plane share weights, the number of free parameters of the network is reduced\cite{5}. Each convolution layer in convolution neural network is closely followed by a computational layer for local average and secondary extraction, which reduces the feature resolution. The special structure of local weight sharing in CNN has unique advantages in speech recognition and image processing. Its layout is closer to the actual biological neural network. Weight sharing reduces the complexity of the network. Especially, multi-dimensional input vector images can be directly input into the network, which avoids feature extraction and classification. The complexity of data reconstruction in the process\cite{6}.

2.1.1 Convolution and Pooling

Convolution is a high-level extraction of local information. Pooling is a special case of convolution. The moving step of pooling is one more than that of convolution. Pooling can greatly reduce the amount of data. Pooling can be divided into maximum pooling and average pooling. Maximum pooling can highlight the key points, while average pooling focuses on local overall characteristics.

The calculation method of convolution:

\begin{equation}
\begin{align*}
1 & \times 1 + 0 \times 2 + 1 \times 3 + 0 \times 0 + 1 \times 2 + 0 \times 0 + 1 \times 1 + 0 \times 2 + 1 \times 3 = 13,
\end{align*}
\end{equation}

The convolution kernel and the calculation of each feature area get the input of the next layer. If the parameters of each feature area are different, then the calculation amount will be huge, so all the parameters of the convolution core are the same by default.

Pooling calculation method (maximum pooling):
Figure 2. The calculation sketch of pooling

Fig. 2 is a sketch of the convolution kernel calculation of 2x2. The step length in the direction of length and width is 2, and each color is the maximum value of the corresponding region. Therefore, the maximum pooling can get the maximum feature of the corresponding region, and can get the maximum feature value of the region image. At the same time, it can be seen that the amount of data pooled has become a quarter.

2.2 Activation Function

There are many activation functions of neural networks, such as relu function and sigmod function (see Fig. 3).

2.3 Prevent Over-fitting

For pages other than the first page, start at the top of the page, and continue in double-column format. The two columns on the last page should be as close to equal length as possible.

Fitting can be understood as taking into account the same value of each data in the model, including errors and some totally worthless data, such as data values that might otherwise be quadratic functions, which are considered as higher-order functions because of their interference and errors. In order to avoid this phenomenon, it is necessary to consider how to prevent over-fitting. The traditional method of reducing over-fitting is dropout, which makes the neurons inactivate randomly during training, so that each neuron does not depend on each other, so as to achieve the method of reducing over-fitting. BN (Batch Normalization) can now be used to standardize batches of BN for one output $wx + b$ at the full connection layer. Firstly, its variance and mean are calculated, $f_c_{\text{var}}, f_c_{\text{mean}}$. Then $\alpha = \frac{wx + b - f_c_{\text{mean}}}{\sqrt{f_c_{\text{var}} + 0.001}}$, in order to ensure that the output can be optimized, the new output is $w \cdot \alpha + b \cdot \alpha$. BN can not only reduce over-fitting, but also keep the overall specifications of training data consistent, thus speeding up training efficiency.

2.4 Softmax

Softmax function mainly normalizes the output probability by expression 6:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$  \hspace{1cm} (1)

j=1,..., K. The expression 1 makes the range of each number in (0,1) interval, and the sum of all numbers is 1. Therefore, it can be considered that the output of the neural network is the probability that a sample belongs to this class. For example, this paper focus on two-class samples, and K is 2.
3. APPLICATION OF CONVOLUTIONAL NEURAL NETWORK IN ASTRONOMICAL IMAGE CLASSIFICATION

3.1 Introduction to Picture Album

There are two types of images used in this paper: one is stars, as shown in Figure 4, and the other is spiral galaxies, as shown in Figure 5. The number of samples is increased by rotation transformation. The following is the case of the two sets.

![Figure 4. Sample maps of spiral galaxies](image1)

![Figure 5. Sample maps of stars](image2)

Both resolutions are 64X64. The first 10,000 samples of spiral galaxies are trained, the last 250 samples are tested, and the first 10,000 samples are trained for stars, and the last 250 samples are tested.

In this paper, we use OS module to import and extract picture files by traversing folders. Then, PIL module is used to store the image pixels in the form of data arrays in the list `img1` and `img2`, and a two-column vector is generated to label them respectively.

3.2 Model Structure Design

In order to reduce the error rate of recognition, the depth of network is added on the basis of CNN. Considering that increasing the depth can improve the non-linearity of the model well, but at the same time there will be over-fitting phenomenon, this paper tries to use dropout and BN to prevent over-fitting in deep learning. BN has achieved good experimental results. As a result, Table 1 is the structure parameter table of the model.
## Table 1. Model structure parameter table

| Input | Details | Output |
|-------|---------|--------|
| Block1 | Conv1_1 | 64 size:3*3 stride:1 pad: SAME | 32*32*64 |
| | Conv1_2 | 64 size:3*3 stride:1 pad: SAME | |
| | Max pooling | stride:2 pad: SAME | |
| Block2 | Conv2_1 | 128 size:3*3 stride:1 pad: SAME | 16*16*128 |
| | Conv2_2 | 128 size:3*3 stride:1 pad: SAME | |
| | Max pooling | stride:2 pad: SAME | |
| Block3 | Conv3_1 | 256 size:3*3 stride:1 pad: SAME | 8*8*256 |
| | Conv3_2 | 256 size:3*3 stride:1 pad: SAME | |
| | Conv3_3 | 256 size:3*3 stride:1 pad: SAME | |
| | Max pooling | stride:2 pad: SAME | |
| Block4 | Conv4_1 | 512 size:3*3 stride:1 pad: SAME | 4*4*256 |
| | Conv4_2 | 512 size:3*3 stride:1 pad: SAME | |
| | Conv4_3 | 512 size:3*3 stride:1 pad: SAME | |
| | Max pooling | stride:2 pad: SAME | |
| Block5 | Conv5_1 | 512 size:3*3 stride:1 pad: SAME | 2*2*256 |
| | Conv5_2 | 512 size:3*3 stride:1 pad: SAME | |
| | Conv5_3 | 512 size:3*3 stride:1 pad: SAME | |
| | Max pooling | stride:2 pad: SAME | |
| Lay1 | Full connect1 | Reshape 2*2*256=1024 (input) | 100 |
| Lay2 | Full connect2 | 100(input) | 80 |
| Lay3 | Full connect3 | 80(input) | 2 |
| Lay4 | BN | 2(input) | 2 |
| Lay5 | softmax | 2(input) | 2 |

The model structure is shown in Figure 6 below:

![Figure 6. Structural chart of improved CNN model](image)

Based on CNN, this paper increases the depth of the model and sets up five blocks altogether. Taking the first Block as an example, continuous convolution is used in the convolution layer to increase the ability of model to extract image information and non-linear fitting. Convolution uses a small convolution core of 3X3 to extract local details and its padding is "SAME". Unlike VALID, SAME can analyze image edge information more effectively by adding 0 in image edge information and convoluting it. It can be seen that the convolution layer of 64 convolution cores changes from the input of 64X64X1 to the output of 64X64X64, which preserves the influence of different convolution cores.
on the data. For the pooling layer, Maxpooling is used instead of Average pooling. The purpose is to enhance the edge information of the image instead of taking the average to achieve smoothing effect. The X direction and Y direction of the string are set to be 2 in order to reduce the data. It can be seen that the output of the pooling layer changes from 64X64X64 to 32X32X64. The last four Blocks are the same, and the last three Blocks add a layer of convolution in order to continue to increase the depth of the model to achieve better training results. Finally, the output data of 2X2X512 are obtained and processed to one-dimensional data into full connection layer. Here we set up three hidden layers and two neurons in the last hidden layer because our classification is a two-dimensional classification, that is, whether it belongs to stars or spiral galaxies. Then adding BN (Batch Normalization), BN can arrange the disordered data in a unified way, which not only accelerates the learning speed but also improves the accuracy for computer learning. In a certain range of the most effective interval of general activation function, BN can integrate the data well in the interval, not only for example. This BN can also reduce the over-fitting, so this paper uses BN. BN standardizes the data in batches, adds weights and biases to the data, which can also be trained to keep the data having the opportunity to change back to its original or better data for training. The traditional method of reducing over-fitting is dropout. Although dropout solves the problem of over-fitting well, there is no excellent effect of BN in the experiment in this paper. If BN is added after dropout, the system error will be increased. It is speculated that BN may produce regularization due to introducing statistics between different samples. The effect replaces the drop out gain. Finally, the function of softmax for classification is added as the output of the whole model.

The processing of BN: (1) The output $wx + b$ is regularized to get the mean value $f_{c,mean}$ and variance $f_{c,var}$ of the whole group. (2) $(wx + b - f_{c,mean})/\sqrt{f_{c,var} + 0.001}$ replaces $wx + b$ for standardized operation. (3) Replacing $x$ with $wx + b$ to pave the way for the next operation. (4) Finally, $x$ is multiplied by the parameter scale (initial 1) and add the translation parameter shift (initial 0), which is used as the output of the layer. These parameters can be trained by the optimizer.

The optimizer used in this paper is Adam algorithm. The general gradient descent method is not a good choice to solve the sparse problem (image processing problem), because the regular term with a norm is not derivable. The Adam algorithm uses not only 1-norm but also 2-norm.

Adam algorithm process: 1. Setting step size. 2. The exponential decay rates of moment estimates, $\rho_1$ and $\rho_2$ are in the interval [0,1]. 3. Small constant $\delta$ for numerical stability. 4. Initial parameter $\theta$. 5. Initialize the first and second moment variables $s = 0, r = 0$. Initialization time step $t = 0$. When $t = 50$, the loop ends. The following are the contents of the cycle: (1) A small batch containing m samples $\{x^{(i)}, \cdots, x^{(m)}\}$ from the training set corresponds to the target of $y^{(i)}$. (2) Computing gradient: $g = \frac{1}{m} \nabla_{\theta} \Sigma_{i} L(f(x^{(i)}, \theta), y^{(i)})$. (3) $t = t + 1$. (4) Updating biased first-order moment estimates: $s = \rho_1 s + (1 - \rho_1) g$. (5) Updating Biased Second Moment Estimation: $r = \rho_2 r + (1 - \rho_2) g \odot g$. (6) Correcting the deviation of the first moment: $\hat{s} = \frac{s}{1 - \rho_1^t}$. (7) Correcting the deviation of the second moment: $\hat{r} = \frac{r}{1 - \rho_2^t}$. (8) Update of calculation: $\Delta \theta = -\epsilon \frac{\hat{s}}{\sqrt{\hat{r} + \delta}}$ (element by element application operation). (9) Application update: $\theta = \theta + \Delta \theta$.

In this image processing, Adam algorithm is superior to Moment, RMSProp and other optimization algorithms. BN instead of traditional dropout has also been verified in this experiment. The activation function chosen in this paper is relu function. The reasons are as follows: 1. Its converge is superior to nce speed is fast. 2. If we use SIGMOD function data to saturate at both ends, it is easy to lose information in the process of propagation, but relu has no problem, tanh has similar problems.

In this paper, 50 image data in each batch are trained and the errors in each batch are output to observe whether the system is effectively trained. Finally, 500 data are used to test the trained model to observe the overall effect of the model. The tensorboard of tensorflow can effectively observe the changes of loss for every 50 data. As shown in Figure 9, the model is stable and tends to be stable in only 60 steps.
Figure 7. Loss change chart

Of course, tensorboard can also output the changes of each parameter in each layer, as shown in Figure 8 for the changes of weights and biases of a fully connected layer:

Figure 8. Partial parameter change diagram of full connection layer

Figure 8 (B) shows that most of the weights are concentrated in the middle (the darker the color, the more concentrated the value), (A) the bias in the middle has become more stable, because the error rate is low, this paper will no longer train.

Finally, the error rate is plotted by Matplotlib module. As shown in Figure 9, although the error rate fluctuates occasionally, the reason is that the image training is a group of 50, each group of samples is too small, but most of the error rates are stable between 0% and 10%. Finally, we tested 500 image sets and found that the error rate was as low as 3%. Figure 10 shows the error rate in terminal.
In this paper, the astronomical images of spiral galaxies and stars are classified based on convolution neural network, and the interference source of pre-star images in spiral galaxy images is eliminated. The model chooses the convolution core of 3X3 to make the information extraction more detailed and deepen the depth of the model. Continuous convolution increases the level of information extraction and enhances the non-linear fitting ability of the model. BN is used to replace the traditional dropout. The full connection layer is followed by the softmax layer, which makes the model a form of probability output. Adam algorithm is used to optimize, and relu is used for all activation functions, which can make the convergence faster. Finally, the visualization results are displayed. The calculation results show that the model has convergence and the error rate is about 3%.

ACKNOWLEDGMENTS
Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

REFERENCES
[1] DE Rumelhart, GE Hinton, RJ Williams, Learning internal representations by error propagation [J]. 1985 – DTIC Document. Fröhlich.
[2] Lecun, Y., Boser, B.E., Denker, J.S. Backpropagation Applied to Handwritten Zip Code [J]. Recognition. Neural Computation, 1989:541-551.
[3] Y.LeCun, L.Bottou, Y.Bengio and P.Haffner. Gradient-based learning applied to document recognition [J], IEEE 86 (11): 2278-2324.
[4] Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning [M]. Beijing: People's Posts and Telecommunications Publishing House. 2017: 187-226.
[5] Yun Ye. Deep Learning and Computer Vision [M]. Beijing: Machinery Industry Press, 2018: 7-90.
[6] Sxron. CNN for Introduction of Neural Networks (2) [Z]. 2018-2-13. DOI=https://www.cnblogs.com/sxron/articles/5850946.html.
[7] Mingnan Chu. Research on Image Classification Technology Based on Convolutional Neural Network [D]. Xiangtan University. 2015.
[8] Xiang Li. Understanding the Disharmony between Dropout and Batch Normalization by Variance Shift [Z]. 2018-1-19. DOI=https://www.zhihu.com/question/265819518/answer/299192.
[9] Wenjian Huang, Yuan Tang. tensorflow actual combat [M]. Beijing: Electronic Industry Press, 2017.2.
[10] Chun Hong Li. Application Research of Vehicle Recognition Platform Based on Structured Image Description [D]. Soochow University. 2016.
[11] Xu Wen. Research and Application of Image Recognition Method Based on Deep Learning [D]. Central China Normal University. 2017.
[12] Baojian Xie. Research on Image Classification Method Based on Convolutional Neural Network [D]. Hefei Polytechnic University. 2015.