Contents lists available at YXpublications

International Journal of Applied Mathematics in Control Engineering

Journal homepage: http://www.ijamce.com

Design of Image Classification Model by Logistic Regression Weighted Fusion Based on Tensor Flow

Yongqiang Zhang*, Yanni Song, Hongbin Gao

College of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050018,China

ARTICLE INFO

Article history:
Received 20 May 2020
Accepted 30 July 2020
Available online 30 July 2020

Keywords:
Neural network
Image classification
Logistic Gaussian
Weighted fusion

ABSTRACT

In order to overcome the difficulty for single network model to simultaneously classify multiple categories of images, prevent misclassification of images with similar feature categories and improve the accuracy of image classification, a image classification model is designed by logistic regression weighted fusion based on TensorFlow. Firstly, image classification of multiple convolutional neural networks is achieved based on TensorFlow. Then a logistic regression weighted fusion model is designed based on each network structure. Our model is constructed by weighted fusion of each neural network classification results. The accuracy and time complexity of the model classification are tested through experiments. Under this data set, the classification accuracy of this model is to a certain extent better than that of using a single network model to classify images. The time complexity has not increased significantly, and the model has achieved good application value.

1. Overview

Today's world is full of colors and various creatures come into view all the time. However, with the continuous enrichment of various types of images and other information, as well as the continuous enhancement of human desire for knowledge, it has become more difficult for humans to quickly identify various types of images in daily life. With the rapid development of search engine technology, and the text search is very mature, but the simple text search is hard to meet the needs of human for the perception of the unknown world. For example, how to obtain the content of the image based on an unknown image and the type of subject matter in the image, so as to realize the classification of the image is particularly important.

In recent years, many scholars have devoted themselves to the study of image classification. As document takes the computer vision object recognition algorithm competition as the main line (e.g., Huang et al., 2014), it has elaborated on the development of the best algorithms for object classification and detection over the years, emphasized the importance of expression learning and structure learning in object classification and object detection. The article also discusses the unity and difference of object classification and detection, considers the development direction of object classification and detection, the image classification is analyzed and prospected from the two directions of expression learning and structure learning based on deep learning.

However, with the rise of deep learning and its superior performance in the field of image processing, more and more researchers have invested in deep learning research. As document reported（e.g., Silver et al., 2016, Li et al., 2018）, the Go program AlphaGo developed by Google's artificial intelligence company DeepMind defeated other Go programs by 99.8% and defeated European professional Go players 5:0. The results achieved by AlphaGo make deep learning a powerful breakthrough in the field of artificial intelligence. In order to accelerate the study of deep learning in all walks of life, as one of the leading companies in the Internet industry, Google has opened up the second-generation machine learning system named TensorFlow. Therefore, with the continuous development of TensorFlow and deep learning, many scholars have accelerated the research on image classification.

Image classification is one of the popular research fields of deep learning, convolutional neural networks are often used to deal with image recognition and other problems（e.g., Affonso et al., 2017, Pan et al., 2017, Gibson et al., 2018, Litjens et al., 2017）. For example, literature (e.g., Chang et al., 2016) summarizes the research progress and typical applications of convolutional neural networks in image understanding, it also expounds the research progress and application of image classification and object detection, face recognition and scene semantic segmentation. Moreover, there are also many research scholars using TensorFlow to classify images (e.g., Rampasek et al., 2016), for example, literature uses TensorFlow to design a video target tracking deep learning model, which used the data in the VOT2015 standard dataset for experimental testing (e.g., Liu et al., 2017), and achieved good accuracy.

* Corresponding author.
E-mail addresses: zyq@hebust.edu.cn (Y. Zhang)

However, because different types of images in the image dataset have certain similarities in terms of color, literary and artistic characteristics, etc. It is easy to cause misclassification when constructing image classification models for a datasets with similar image categories, which affects the classification accuracy. As a depth information framework, TensorFlow has excellent performance in image classification, but the classification result is directly related to the selected algorithm and network structure. Although the convolutional neural network has superior performance in image classification, it is disturbed by factors such as different image with different image information, which causes different network structures have different advantages and disadvantages and performance in different types of images, and it is difficult to design a general network structure which has good performance in various categories of images. Therefore, in order to improve the accuracy of classifying images in various categories, this paper designs an image classification model based on TensorFlow and logistic regression to complete the classification of images.

The image classification model based on the weighted fusion of TensorFlow and logistic regression designed in this paper first based on TensorFlow which uses VGG, AlexNet, ResNet and other better networks to achieve the classification of images, and then, based on the classification results of multiple networks, a logistic regression model is used to perform weighted fusion on the classification results of each network, and finally realize the classification of images. This paper tests the model through experiments, and the results show that the model can improve the accuracy of image classification under the dataset used in this paper.

2. Design of image classification model

2.1 TensorFlow image classification model design

In order to improve the accuracy of image classification, this paper uses TensorFlow to design an image classification model, using VGG, AlexNet, ResNet and other better networks to achieve the classification of images.

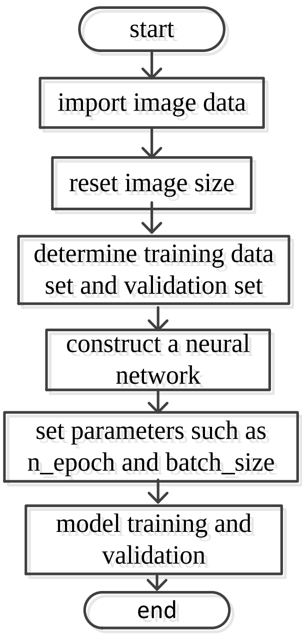

Fig.1. Flowchart of image classification model training based on TensorFlow

When constructing an image classification model, the original input image data needs to be set to a uniform size, and then the datasets are divided into train set and the test set. The core link of the model construction lies in the design of the neural network structure, in the process of designing the network structure, you need to input the 2D convolution function tf.nn.conv2d()in tensorflow, the maximum pooling function tf.nn.max_pool(), tensor's deformation function tf.reshape(), other parameters of the neural network (e.g., Wen et al., 2018, Liu et al., 2018), etc, moreover, the ReLU function was used as the activation function for nonlinear processing during the model construction (e.g., Gu et al., 2018, Tang et al., 2017, Li et al., 2017, Paoletti et al., 2017). This paper uses TensorFlow to construct the image training model of the image classification model as shown in Figure 1.

The pseudocode of the convolutional neural network structure design when using TensorFlow to build an image classification model is shown in Table 1.

Tab.1. The pseudo code of neural network structure design

input	tf.placeholder（）
output	logit
	input tf.placeholder（）
	determine the values of parameters m and n
	for i=1,2,…,m do
	define conv[i]_weights = tf.get_variable()
	define conv[i]_biases = tf.get_variable()
	define the 2D convolution function conv [i] = tf.nn.conv2d ()
	use the relu activation function to do the Nonlinear processing：relu[i]=tf.nn.relu(tf.nn.bias_add(conv[i], conv[i]_biases))
	define the maximum pooling function pool[i]= tf.nn.max_pool(
	end for
	convert image structure information by reshaped = tf.reshape ()
process	for j=1,2,…,n do
	define fc1_weights = tf.get_variable()
	if regularizer != None then
	define tf.add_to_collection('losses', regularizer(fc[i]_weights
	else
	save the current state without doing anything
	end if
	define fc1_biases = tf.get_variable()
	if the layer is a non-output layer then
	fc[i]=tf.nn.relu(tf.matmul(reshaped,fc[i]_weights)+fc[i]_biase
	if executes this layer of network then
	Dropout layer fc[i] = tf.nn.dropout(fc[i], p) （0<p<1）
	else
	save the current state without doing anything
	end if
	else
	calculate the result logit = tf.matmul(fc[i-1], fc[i]_weights) + fc[i]_biases
	end if
	end for

Since the neural network to be executed will have many weights, offsets and other parameters to be created, when designing the structure of the neural network, it is necessary to define some initialization functions and create some random noise for the weights to break the original symmetrical structure, such as setting the standard deviation, etc. In the process of designing this network structure, this paper uses the ReLU function as the activation function, therefore, it also adds some values to the offset to achieve the purpose of avoiding dead nodes (e.g., Yarotsky et al., 2017). And through the tf.nn.dropout() function introduced the dropout layer (e.g., Cao et al., 2018, Tang et al., 2017, Peng et al., 2018). In the image classification model designed in this paper, after the dataset is input to the network through the input layer, a series of processing and operations in the above steps, the image classification results are output by output layer finally.

2.2 VGG, AlexNet and ResNet image classification model design

In order to obtain better classification results, based on the above-mentioned image classification model, this paper uses VGG, AlexNet, ResNet and other better networks to classify images.

VGGNet is used to extract image features, and it is composed of five layers of convolutional layers, three layers of fully connected layers, and softmax output layers. The layers are separated by max-pooling, and the activation units of all hidden layers are used ReLU function. Its outstanding contribution is to prove that using a small convolution and increasing the network depth can effectively improve the effect of the model, and VGGNet has a good generalization ability for other datasets. VGGNet explores the relationship between the depth of CNN and its performance. The deeper the network, the better. However, the training of VGG networks is very slow, and due to the depth and the fully connected layer at the end, they require a larger weight storage space.

AlexNet has one input from left to right, and then there are eight layers that need to be trained, the first five are convolutional layers, and the last three layers are fully connected layers. We use Dropout to randomly ignore some neurons during training to avoid overfitting of the model. It is generally used in the fully connected layer. Dropout is not used in prediction, that is, Dropout is 1. We use overlapped maximum pooling in CNN (step size is smaller than convolution kernel). Previously, average pooling was commonly used in CNN, and the use of maximum pooling can avoid the blurring effect of average pooling. At the same time, overlapping effects can increase the richness of features.

AlexNet uses ReLU instead of Sigmoid, which can train faster, and it solves the vanishing gradient of sigmoid in a deeper network at the same time. However, the value range obtained by the ReLU activation function does not have a range, so the results obtained by ReLU must be normalized, which is Local Response Normalization. It can create a competition mechanism for the activity of local neurons, which makes the value with larger response become relatively larger, and inhibits other neurons with smaller feedback, which enhances the generalization ability of the model. The method is as formula (1).

$$b^i_{(x,y)} = \frac{a^i_{(x,y)}}{(k+\alpha\sum\limits_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)}(a^i_{(x,y)})^2)^\beta} \tag{1}$$

Among them, $a^i_{(x,y)}$ represents the output of ReLU at the position (x, y) of the i-th kernel, $b^i_{(x,y)}$ represents the result of LRN, n is the number of adjacent kernel maps at the same position, and N is the total number of kernals.

The result of ReLU output and its neighbors in a certain range are locally normalized, which is a bit similar to our maximum and minimum normalization.We Suppose there is a vector such as expression(2), then the normalization rule of normalizing all numbers to 0-1 can be expressed by formula(3)：

$$X = [x_{1,} x_2, ..., x_n] \tag{2}$$

$$x_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{3}$$

The entire AlexNet has eight layers that require training parameters. LRN appears after the first and second convolutional layers, and the max pool appears after the two LRN layers and the last convolutional layer. And ReLU all appear behind each of these eight layers. The parameter calculation formula is as follows.

$$W_{out} = \frac{W_{out} - size + 2\times padding}{stride} + 1 \tag{4}$$

$$H_{out} = \frac{H_{out} - size + 2\times padding}{stride} + 1 \tag{5}$$

$$D_{out} = output\, depths \tag{6}$$

ResNet is also known as Deep Residual Learning. Deep networks are difficult to train. ResNet solves the problems of gradient disappearance and gradient explosion, and extends the network depth to a maximum of 152 layers. The formula for Residual Learning is as formula (7),

$$H(X) = F(X) + X \tag{7}$$

In the ordinary convolution process, an identity mapping of X is added, assuming that the residual is H(X), in the Residual Learning, you can learn F(X) first, and then add X, it can greatly improve performance, and then cross-layer connections are made through summation. The Residual Learning structure is shown in Figure 2.

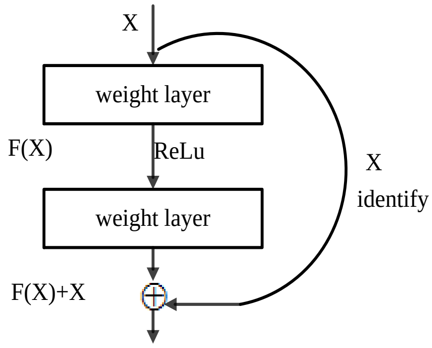

Fig.2. Residual Learning

The Residual structure can be simply written as the following structure, as shown in formula (8), through recursion and formula (9), this shows that there are residual characteristics between any unit L and l, the X_L is the product of a series of matrix vectors. For back propagation, assuming that the loss function is E, according to the chain rule of back propagation, we can get function (10), this formula also ensures that the gradient will not disappear.

$$x_{l+1} = x_l + F(x_l, W_l) \tag{8}$$

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i, W_i) \tag{9}$$

$$\frac{\partial \varepsilon}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}(1 + \frac{\partial}{\partial x_l}\sum_{i=l}^{L-1} F(x_{i,} w_i)) \tag{10}$$

In formula (10), we can find that this derivative is divided into two parts, which are not passed through the weight layer, such as the following expression (11). What is passed through the weight layer is shown as formula(12).

$$\frac{\partial \varepsilon}{\partial x_L} \tag{11}$$

$$\frac{\partial \varepsilon}{\partial x_L}(\frac{\partial}{\partial x_l}\sum_{i=l}^{L-1} F(x_i, w_i)) \tag{12}$$

Between the two expressions, the former ensures that the signal can be directly transmitted back to any shallow such as x_l, and because the formula(13) cannot be -1, so it avoids the vanishing gradient.

$$\frac{\partial}{\partial x_l}\sum_{i=l}^{L-1} F(x_i, w_i) \tag{13}$$

2.3 *Logistic regression model design*

Logistic Regression model is a classification model, expressed in the form of conditional probability distribution as $P(X \mid Y)$, random variable X takes value of n-dimensional real number vector, such as $X = (X^{(1)}, X^{(2)}, \ldots X^{(n)})$, Y takes the value 0 or 1, the formula is as follows.

$$P(Y=1 \mid X) = \frac{\exp(\omega.x+b)}{1+\exp(\omega.x+b)} \tag{14}$$

$$P(Y=0 \mid 0) = \frac{1}{1+\exp(\omega.x+b)} \tag{15}$$

$$\phi(x) = \frac{1}{1+e^{-\omega^T x - b}} \tag{16}$$

The predicted value produced by the linear regression model is z, the expression as formula (17). Where z is a real value, since the function is discontinuous. We find the Sigmoid function instead, the function is shown in formula (18),

$$z = \omega^T x + b \tag{17}$$

$$\phi(z) = \frac{1}{1+e^{-z}} \tag{18}$$

In addition, the Sigmoid function has a very good property, the formula (19) is expressed as follows. The Sigmoid function image is shown in the figure below.

$$\phi'(z) = \phi(z)(1-\phi(z)) \tag{19}$$

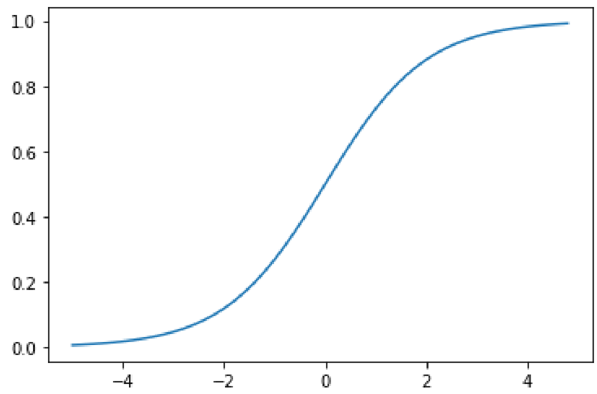

Fig.3. Sigmoid function

With Sigmoid function, since its value is in [0,1], we can regard it as the posterior probability estimate p(y=1|x) of class 1. To put it bluntly, if there is a test point x, then the result calculated by Sigmoid function can be used as the probability that the point x belongs to category 1.

So, very naturally, we classify the value calculated by Sigmoid function greater than or equal to 0.5 as category 1, and the value less than 0.5 as category 0, the formula (20) as follows.

$$\hat{y} = \begin{cases} 1 & if\, \phi(z) \ge 0.5 \\ 0 & otherwise \end{cases} \tag{20}$$

At the same time, logistic regression and adaptive linear network are very similar. The difference between the two is that the activation function of logistic regression is Sigmoid function and the activation function of adaptive linear network is y=x. The network structure of the two is shown in the figure below (e.g., Raschka et al., 2017).

The network structure of the logistic regression network is shown in the figure below.

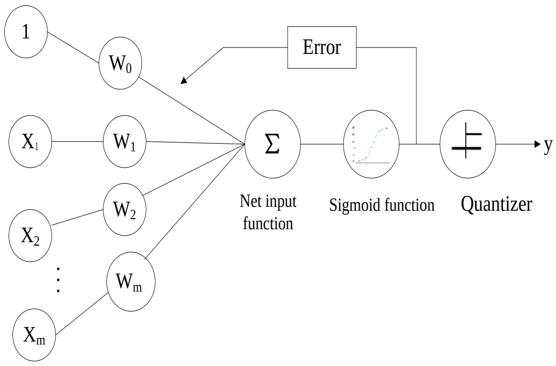

Fig.4. Logistic regression network

The network structure of the adaptive linear network is shown in the figure below.

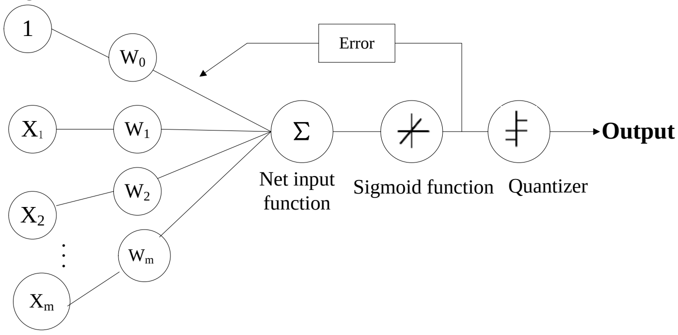

Fig.5. adaptive linear network

To find the parameter w, you must first define the cost function, which is the objective function. Imitate the approach of linear regression, using the error sum of squares as the cost function. The formula (21) is as follows.

$$J(w) = \sum_{i=1}^{n} \frac{1}{2}(\phi(z_i) - y_i)^2 \tag{21}$$

In the formula (21), y_i is the real class label of X_i, and the expression of z_i is shown in formula (22).

$$z_i = W^T X_i + w_0 \tag{22}$$

The current cost function image is a non-convex function, and the non-convex function has more than one extreme point, which makes it difficult to perform optimization calculations. Then we express the new cost function by the following formula (23).

$$p(y \mid X; W) = \phi(z)^y (1-\phi(z))^{1-y} \tag{23}$$

At this time, the joint probability formula (24) is as follows.

$$\prod_{i=1}^{n} p(y_i \mid X_i; W) \tag{24}$$

For this joint probability, we can determine the parameter W that maximizes the joint probability by calculating the maximum likelihood estimation method of the parameters. At this time, W is the best parameter we want to choose, which maximizes the joint probability, that is, the cost Minimal function.

To calculate logistic regression, we must first obtain the maximum likelihood function, the derivation process formula (25) is as follows.

$$\begin{aligned} L(W) &= \sum_{i=1}^{n} \ln p(y_i \mid X_i; W) \\ &= \sum_{i=1}^{n} \ln(\phi(z_i)^{y_i}(1-\phi(z_i))^{1-y_i}) \\ &= \sum_{i=1}^{n} y_i \ln\phi(z_i) + (1-y_i)\ln(1-\phi(z_i)) \end{aligned} \tag{25}$$

Through analysis, we get -L(W) is J(W), the formula (26) is as follows.

$$J(W) = -\sum_{i=1}^{n} y_i \ln\phi(z_i) + (1-y_i)\ln(1-\phi(z_i)) \tag{26}$$

J(W) is a non-linear sigmoid function, which cannot be solved directly by the partial derivative of 0. So we use the gradient descent method.

First of all, according to the related theory of gradient, we know that the negative direction of the gradient is the direction in which the cost function drops the fastest. Therefore, we should gradually adjust the weight component w_j along the negative direction of the gradient until the minimum value is obtained, so the change of each weight component should be like this formula (27).

$$\Delta w_j = -\eta \frac{\partial J(W)}{\partial w_j} \tag{27}$$

Among them, η is the learning rate and it used to control step length, the formula (27) also can be calculated as follows formula (28). At this point, we have the variable for updating the weight of the gradient descent method, it is shown as the formula (29).

When the sample size is very large, updating the weights each time will be very time-consuming. In this case, the stochastic gradient descent method can be used. At this time, the sample needs to be shuffled at each iteration, and then the weights are continuously updated with the following formula (30).

$$\begin{aligned} \frac{\partial J(W)}{\partial w_j} &= -\sum_{i=1}^{n}(y_i \frac{1}{\phi(z_i)} - (1-y_i)\frac{1}{1-\phi(z_i)}).\frac{\partial\phi(z_i)}{\partial w_j} \\ &= -\sum_{i=1}^{n}(y_i(1-\phi(z_i)) - (1-y_i)\frac{1}{1-\phi(z_i)}).x_{ij} \\ &= -\sum_{i=1}^{n}(y_i - \phi(z_i)).x_{ij} \end{aligned} \tag{28}$$

$$w_j = w_j + \eta\sum_{i=1}^{n}(y_i - \phi(z_i)).x_{ij} \tag{29}$$

$$\mathrm{w}_j = w_j + \eta(y^i - \phi(z^i))x_j^{(i)} \tag{30}$$

2.4 Logistic regression weighted fusion model design

TensorFlow programs are usually organized into a construction phase and an execution phase. In the construction phase, graphs are used to represent computing tasks, and the nodes in the graph are called op (abbreviation for operation). In the execution phase, use the op in the session execution graph. At the same time, TensorFlow converts the graph definition into distributed execution operations, and provides a method to execute op to make full use of available computing resources.

TensorFlow uses tensor to represent data. An op can get one or more tensors, perform calculations, and generate one or more tensors. Each tensor is a typed multidimensional array. In the execution stage, after the op method is executed, the generated tensor is returned

There are many weighted mixed models, in addition to simple linear models, Logistic Regression, Restricted Boltzmann Machines and Gradient Boosted Decision Trees are commonly used.

The main steps of training through TensorFlow are: preparing data, building models, training models, and making predictions.

When using tensorflow to implement a simple logistic regression algorithm, the logistic regression can be regarded as a forward neural network with only one layer of network, and the weight of the parameter connection is only a value, not a matrix. The formula is as the formula (31), where X is the input, W is the weight between the input and the hidden layer, b is the bias of the hidden layer neuron, and logistic is the activation function, generally sigmoid or tanh, y_predict is the final prediction result..The objective function is the L2 distance between y_predict and the real label Y, and the stochastic gradient descent algorithm is used to update the weights and biases.

$$y\_predict = \log istiic(X * W + b) \tag{31}$$

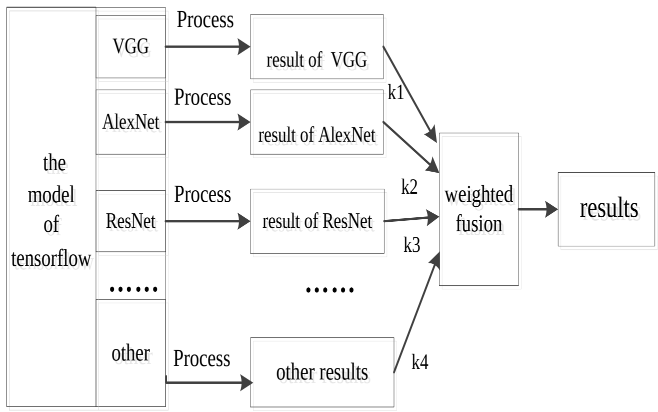

Fig.6. Technical roadmap of logical regression weighted fusion model design

This article designs the TensorFlow image classification model in section 2.1. Based on the TensorFlow image classification model designed in section 2.1, various neural networks such as VGG, AlexNet, and ResNet are used to train the image classification model, and different classification results are obtained. However, because different network results have their own advantages and disadvantages in different applications, in order to obtain more accurate classification results, this section uses logistic regression to design an image classification weighted fusion model. The design roadmap for this model is shown in Figure 6. Through this figure, we can get the design concept of the weighted fusion model. This figure intuitively expresses the model composition and design ideas of the weighted fusion model

In section 2.1 of this paper, firstly we use a variety of neural networks to achieve the classification of images, for the classification results of each network, we need to replace the classification results of each image for each category with a form of 0 or 1. Among them, 1 represents the classification result of the corresponding image by the network as this category, and 0 represents other categories. For example, if the classification result of a certain image by a certain network is class A, the classification result of this image by the network is converted into a form where class A is 1, and other classes are 0.

After the training of the image classification model of all network structures is completed, based on the classification results of multiple networks, a weighted fusion is performed using a logistic regression model. That is, for each image category, the classification result (0 or 1) of the network model after conversion relative to the category is used as the input of the logistic regression model, and the real category result (0 or 1) is used as the label of the result for logistic regression, after the model training, the weighted fusion coefficients such as $k_1, k_2, ..., k_n$ of each network model with respect to each image category are obtained. Then based on the classification results and weighted fusion coefficients of multiple networks, for each image category, use the formula to calculate the classification result coefficient Y of each image relative to each category, and select the category with the largest classification coefficient Y as the classification result of the image. In formula (33), x_i represents the classification result (0 or 1) of the network model relative to the corresponding category conversion, k_i is the corresponding weighted fusion coefficient, which requires $0 < k_i < 1$, and it needs to satisfy the expression (32).

$$\sum_{i=1}^{n} \mathrm{k}_i = 1 \tag{32}$$

$$Y = \sum_{i=1}^{n} k_i x_i \tag{33}$$

In the weighted fusion designed in this paper, the weighted fusion coefficients of each network model need to be calculated for each category label in the dataset such as $k_1, k_2, ..., k_n$. For the weighted fusion coefficients of different categories, we use formula (33) to calculate the classification coefficient Y of the corresponding category, and then according to the calculation result of the classification coefficient Y of different categories, the category corresponding to the largest classification coefficient Y is selected as the final classification result of the weighted fusion image classification model in this paper.

3. Experimental plan design and result analysis

3.1 Experimental data selection and experimental plan design

This article sets the third parameter in weights to three channels when using the TensorFlow two-dimensional convolution function tf.nn.conv2d (input, weights, strides, padding). That is, using color images as datasets for model training and test (e.g., Denisova et al., 2017, Tzelepi et al.,2017). Therefore, the dataset in this paper needs to use color images. First, the experiment in this paper used images containing eight types of animals to train and test the model, among them, 70% of the dataset was selected as the training set, and the remaining 30% of the dataset was used as the test set. An example of some images in this dataset is shown in Figure 7. This figure shows an image from each type of image in all categories in this dataset.

In the experiment of this paper, the dataset used in this experiment trained and tested through the four network structures. The image classification accuracy of each network model and weighted fusion model in the dataset used in this experiment is shown in Figure 8.

Fig.7. Partial datasets examples

In this paper, the image classification model is trained by a variety of better networks such as VGG, AlexNet and ResNet. Then based on the classification results of each network model, the final image classification model is trained and tested using the weighted logistic regression fusion model, and the final result of image classification is obtained.

3.2 Comparison and analysis of experimental results

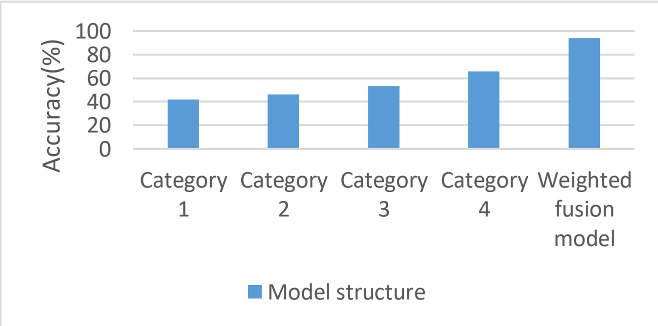

Fig. 8. Comparison of experimental results

Figure 8 shows the superiority of the weighted fusion model in image classification. The accuracy rate is higher than that of the four network models used in the experiments in this paper for image classification to a certain extent. This shows that the model in this paper has certain usability and superiority.

3.3 Experimental comparison and analysis of model time complexity

In order to improve the usability of the model, this paper compares and analyzes the time complexity of the model. When comparing the time complexity experiments in this paper, because the image classification time is directly related to the experimental hardware environment, experimental data, network structure and parameters, etc. Therefore, this paper compares the execution time of the weighted fusion model and each single model of the same dataset under the same experimental environment during the experiment.

Tab.2 . Time comparison for weighted fusion model

number	comparison of the shortest time model	comparison of the longest model
1	1.88	1.30
2	1.67	1.19
3	1.25	1.13
4	1.53	1.21
5	2.11	1.30

In the experiment, this paper uses multiple identical machines to carry out the experiment, that is, each machine runs each single image classification model separately, and then the single model image classification results of each machine are weighted and fused. In this experiment, the working time of each single model and the working time of the weighted fusion model under various experimental parameters and datasets are respectively counted. It's shown in table2.

Table 2 shows the multiples of the shortest running time and the longest running time between the weighted fusion model and each single model in the same experimental environment. It can be seen from Table 2 that the weighted fusion model in this paper does not cause a significant increase in time in terms of time complexity, so the model has strong usability.

4. Conclusion

Image classification is a popular research field in deep learning. Since the computer needs to convert image information into binary data during image processing, the accuracy of image classification will be disturbed to a large extent by image color, arts and literature and other characteristics. Therefore, in order to reduce the phenomenon of misclassification of similar images in the image classification process, this paper uses TensorFlow and logistic regression to construct a weighted fusion image classification model. The model first uses a variety of network structures to achieve image classification. Based on the classification results of each network structure, a weighted fusion model is constructed using logistic regression to achieve image classification. The experimental test results show that the accuracy of the model for image classification is better than the single network model under the dataset of this paper, and the time complexity has not caused a significant improvement. However, because this paper only outputs two forms of 0 and 1 when converting the image classification results of each network structure, the weighted fusion result may be affected to some extent. Therefore, in the follow-up work, the author intends to output a score in the [0, 1] interval as the input of the weighted fusion model, which will have a positive effect on the accuracy of image classification. However, in general, the model has certain application value. The model can be used to classify and retrieve images, and the accuracy of classification has been improved to a certain extent.

Acknowledgements

The work has been partly supported by Natural Science Foundation of Hebei Province (F2018208116); Hebei Science and Technology Support Plan Project (16210312D); Soft Science Funding Project of Shijiazhuang Science and Technology Bureau (195790055A).

References

Huang K.Q., Ren W.Q., Tan T.N., 2014. A survey of image object classification and detection algorithms. Journal of Computer, 36(6): 1225-1240.

Silver D., Huang A., Maddision C.J., et al, 2016. Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587):484-489.

Li, S. and L. Chen, 2018. Research on Object Detection Algorithm Based on Deep Learning. International Journal of Applied Mathematics in Control Engineering, 1(2): p. 127-135.

Affonso, C., Debiaso Rossi, A.L., Antunes Vieira, F.H., 2017. Deep learning for biological image classification. Expert Systems with Applications, (85): 114-122.

Pan B., Shi Z.W., Xu X., 2017. MugNet: Deep learning for hyperspectral image classification using limited samples. Isprs Journal of Photogrammetry & Remote Sensing.

Gibson, E., Li, W.Q., Sudre, C., et al, 2018. Niftynet: a deep-learning platform for medical imaging. Computer Methods and Programs in Biomedicine, (158): 113-122.

Litjens, G., Kooi, T., Bejnordi, B. E., et al, 2017. A survey on deep learning in medical image analysis. Medical Image Analysis, 42(9), 60-88.

Chang L., Deng X.M., Zhou M.Q., et al, 2016. Convolutional neural network in image understanding. Journal of Automation., 42(9): 1300-1312.

Rampasek L., Goldenberg A., 2016. TensorFlow: Biology's Gateway to Deep Learning?. Cell Systems,2(1):12-14.

Liu F., Liu P.Y., Li B., et al, 2017. Deep learning model design of video target tarcking based on TensorFlow platform. Laser & Optoelectronics, 54(9): 277-285.

Wen, S., D. Liu, and X. Wang, 2018. Finite-Time Stability of Fractional Order Neural Networks with Proportional Delays. International Journal of Applied Mathematics in Control Engineering, 1(2): p. 111-118.

Liu, Y., et al., 2018. Modified CRM-based Model Reference Adaptive Control with Reduced Peaking Phenomenon. International Journal of Applied Mathematics in Control Engineering, 2018. 1(1): p. 70-78.

Gu J.X., Wang Z.H., et al, 2018. Recent advances in convolutional neural networks. Pattern Recognition, (77): 354-377.

Tang H., Xiao B., Li W.S., et al, 2017. Pixel Convolutional Neural Network for Multi-Focus Image Fusion. Information Sciences. (433): 125-141.

Li S.M., Fan R., et al, 2017. A Two-channel Convolutional Neural Network for Image Super-Resolution, (275): 267-277.

Paoletti, M.E., Haut, J.M., Plaza, J.,et al, 2017. A new deep convolutional neural network for fast hyperspectral image classification. Isprs Journal of Photogrammetry & Remote Sensing.

Yarotsky., Dmitry., 2017. Error bounds for approximations with deep relu networks. Neural networks, (94):103-114.

Cao J.L., Pang Y.W., Li X.L., et al, 2018. Randomly Translational Activation Inspired by the Input Distributions of ReLU. Neurocomputing, (273): 859-868.

Tang Z.M., Luo L.K., Peng H., et al, 2017. A Joint Residual Network with Paired ReLUs activation for Image Super-Resolution. Neurocomputing. (273): 37-46.

Peng Y.L., Zhang L., Zhang Y., et al, 2018. Deep deconvolution neural network for image super resolution. Journal of Software, 29(4): 1-9.

Raschka S., 2015. Python Machine Learning. Packt Publishing.

Denisova, A.Y., Sergeyev, V.V., et al, 2017. Supervised multichannel image classification algorithm using hierarchical histogram representation. Procedia Engineering, (201):213-222.

Tzelepi, M., Tefas, A., 2017. Deep convolutional learning for content based image retrieval. Neurocomputing, (275): 2467-2478.

Zhang Yongqiang born in 1981, is a PhD candidate, an associate professor, and a master's supervisor. He was graduated from Anhui University of Technology and received master's degree of Computer Application Technology. Main research areas: artificial intelligence, Internet of things technology.

Song Yanni born in 1994, graduated from Hebei University of Science and Technology with a major in Computer Technology. The main research directions are: digital image processing technology, artificial intelligence.

Gao Hongbin born in 1964, professor, and a master's supervisor. His main research

directions are: Computer measurement and control, Internet of Things.