The Optimization of Multi-classifier Ensemble Method Based on Dynamic Weighted Voting

Ping Yang¹, Jian Fang², Junting Xu³, Guanghao Jin⁴ and Qingzeng Song³*

¹School of Software, Tian Gong University, Tianjin, China
²Beijing Gongmei Group, Beijing, China
³School of Computer Science and Technology, Tian Gong University, Tianjin, China
⁴School of Telecommunication Engineering, Beijing Polytechnic, Beijing, China
*Email: qingzengsong@tiangong.edu.cn

Abstract. Generally, on the same data set, different deep learning classification models will achieve different performances. The existing weighted voting method can combine the results of models, which can improve the performance of classification. On the other side, its classification accuracy is affected by the accuracy of all models. In this paper, we proposed a dynamic weighted voting method. Our method dynamically selects models on different data sets, and integrates them according to their weights, thereby improving the classification accuracy. We evaluated the methods on three data sets of CIFAR10, CIFAR100 and Existing, which increased the accuracy about 0.65%, 0.91%, and 0.78% respectively compared with the existing weighted voting method.

1. Introduction
Generally, different deep learning classification models have strong or weak ability to obtain the features on the same data set. At the same time, the prediction results of these classifiers are complementary in some samples. By integrating multiple classifiers, the overall classification performance can be improved. The existing weighted voting method is based on the output information of each classifier, which combines the results of models based on the weight of these classifiers. On the other side, some classifiers with poor classification performance may reduce the accuracy. To solve this problem, this paper proposed a new ensemble method based on the existing weighted voting ensemble method. This method is mainly improved the performance by the following two aspects: we firstly expand the consideration set of models, and then we dynamically select the proper number of models based on the validation set.

The structure of this paper is organized as the following: the first section introduces the background of this paper. The second section introduces related works. The third one introduces our method. The fourth section describes and analyses the experimental results. Finally, the fifth section summarizes this paper and introduces the future work.

2. Related Works
In recent years, the research of computer becomes hot with the utilization of deep learning [1-4]. Image classification has also attracted a large number of researchers, which is widely used in various fields, such as medical image segmentation [5], face recognition [6] and remote sensing image classification [7]. Lecun et al proposed the lenet model [8] and achieved good results in handwritten numeral recognition in minist data set. This model takes the picture as the input, extracts the feature
information in the image through the stacked convolution layer and pooling layer as the feature extractor, and finally classifies and outputs the image through the full connection layer and activation function. Although CNN (convolutional neural network) [9] models continuously improves the classification accuracy, the ability of a single model is still limited. The accuracy depends on many factors like the size of figures or the gradient vanishing problem [10]. Ensemble methods [11] try to combine multiple models with certain strategies to improve the overall classification performance.

Voting methods [12-13] are to combine classifiers by using the idea of ensemble learning. Different from the simple voting method, the weighted voting method [14] gives different weights to different classifiers, and finally obtains the results. Weighted voting method is widely used in various aspects. For example, Mu X et al. [15] propose the application of weighted voting methods to face recognition and speech recognition, and Sammar Moustafa et al. [16] use weighted voting to predict software bug. The weight allocation method has a certain impact on the overall performance. How to accurately allocate large weight to the base classifier with large contribution is also a key problem. Alican Dogan [17] proposed a new weight allocation method from the perspective of weight allocation. Different from the existing literature, this method only provides the reward mechanism. In the validation set, the weight of the base classifier with correct instance classification is increased by the percentage of other base classifiers with incorrect instance classification. However, the disadvantage of this method is that it combines all the base classifiers. Although it makes use of the complementary information between the base classifiers to a certain extent, integrating all the base classifiers greatly affected by the accuracy of each classifier. Even if the weight of the base classifier with poor performance is small, it still has a certain impact on the overall classification performance. Therefore, how to select the most complementary base classifiers among the existing base classifiers for integration is important. Erdal Tasici [18] et al. Proposed to generate a series of variant models based on different image enhancement and pre-processing. Among all of the models, this method selects some of the best models for weighted voting integration. This method has certain advantages over the latest results in the detection of tuberculosis.

In this paper, we try to dynamically select the trained models. Compared with the existing method that only applies the voting method to the results of models, we try to combine the outputs of these models that are the probability of labels to achieve higher accuracy. Furthermore, the number of models for the combination is also dynamically selected based on the validation set, which is to reduce the effect of the low accurate models.

3. Materials and Method

The framework of our method is shown in figure 1. Let the sample set s be $[s_1, \ldots, s_n]$, the label set l be $[l_1, \ldots, l_n]$, the ground truth set G be $[g_1, \ldots, g_n]$, and the classifier set C be $[c_1, \ldots, c_n]$, among them, the classifier set C is an ordered set, which is sorted according to the accuracy obtained on the validation set. $p_{i,k}$ is the probability value labeled $l_k$ given by the i-th base classifier for a sample s. For each sample, the output of the i-th base classifier is $p_i = [p_{i,\text{max}_1}, \ldots, p_{i,\text{max}_j}, \ldots, p_{i,\text{max}_k}]$, where $p_{i,\text{max}_j}$ is the probability value after sorting according to the label probability value, and the corresponding label is marked as $l_{i,\text{max}_j}$.

Our method trains each base classifier on the training set, runs the trained classifier on the validation set, and obtains different weights $\omega$ [17], i.e. $[\omega_1, \ldots, \omega_j, \ldots, \omega_m]$, through continuous iterative updating, where $\omega_j$ is the weight of the j-th base classifier. At the same time, the base classifier is sorted according to the accuracy obtained from the validation set to obtain the ordered classifier set C.
For any sample $s$, the probability distribution $P = \begin{bmatrix} p_{1,\text{max},1} & \cdots & p_{1,\text{max},k} \\ \vdots & \ddots & \vdots \\ p_{m,\text{max},1} & \cdots & p_{m,\text{max},k} \end{bmatrix}$ can be obtained. For the $i$-th classifier, the first $\hat{c}_i$ labels with the best probability value are selected. The value of $\hat{c}_i$ is given according to $c_i$ the classification accuracy. For example, if $c_1$ has the best classification accuracy, then $\hat{c}_1$ is $m$, the selected probability subset is $\{p_{1,\text{max},1}, \ldots, p_{1,\text{max},m}\}$, and the corresponding label subset is $\{l_{1,\text{max},1}, \ldots, l_{1,\text{max},m}\}$. If the classification accuracy of $c_m$ is the worst, $\hat{c}_m$ is 1. The selected probability subset is $\{p_{m,\text{max},1}\}$ and the corresponding label subset is $\{l_{m,\text{max},1}\}$. From the probability subset and the label subset, the probability set $\hat{P}$ and the corresponding label set $\hat{L}$ can be obtained respectively. The result candidate set $D = \{d_{i,\text{max},j}\}$ can be obtained from formula (1). Where $d_{i,\text{max},j}$ is obtained from equation (1).

$$d_{i,\text{max},j} = p_{i,\text{max},j} \times q_i, p_{i,\text{max},j} \in \hat{P}$$

**Figure 1.** The framework of our method.

At the final step, the label with the highest confidence is taken as the final decision of this method. In addition, we also consider selecting a certain number of best classifiers for integration. The number of optimal base classifiers $n$ is obtained by the validation set.

4. Experiment and Analysis

4.1. Data Processing

In order to prove the effectiveness of the proposed method, three public data sets cifar10, cifar100 and Existing are selected and compared with the existing weighted voting method. The three data sets are described in table 1.
Table 1. Three datasets used in the experiment.

| Dataset name | Number of samples | Number of classes | Size of training set |
|--------------|-------------------|------------------|----------------------|
| CIFAR10      | 60000             | 10               | 6000                 |
| CIFAR100     | 60000             | 100              | 600                  |
| Existing     | 60000             | 100              | 600                  |

The size of the training set is shown in Table 1. The cifar10 dataset has 60000 instances in total and 10 classes, with 6000 instances in each class. 5000 instances are randomly divided from each class, and a total of 50000 instances are used as the training set to train the base classifier. The remaining 10000 instances are used as validation sets and test sets. cifar100 is similar to cifar10, except that cifar100 has 10 categories and 600 instances of each category. Each class is divided into 500 instances as the training set and the rest as the validation set and test set. The Existing dataset has 100 classes, 600 instances in each class, 480 instances in each class as the training set, and the rest as the validation set and test set. The images are normalized to facilitate better input into the network for training. In the training process, the data enhancement method is adopted for the pictures, and the data set is further expanded by randomly rotating the pictures horizontally to prevent over fitting and enhance the robustness of the network.

4.2. Experimental Process
In this paper, six convolutional neural networks, vovnet [19], vgg [20], res2net [21], repvgg [22], resnet [23] and densenet [24], are used as the base classifiers. Firstly, the picture is input as the size of 256 * 256. Models extract the features of the picture through a series of convolution layers and pooling layers. Finally, the picture is classified through the full connection layer, and the probability vector is output, that is, the probability of the picture corresponding to each label. We also use the pre training model based on Imagenet training. The advantage of this is to use the pretraining model trained on a large data set to load some general features, and train on this basis, which can greatly save the training cost. The mini batch gradient descent method is adopted, that is, the training samples are divided into multiple batches of the same size for training, which not only updates the weight, but also improves the convergence speed. At the same time, in order to better process the data, we deal with the probability distribution in the form of matrix.

The experimental process is shown in figure 2. By running the trained base classifier on the validation set, the weight of the base classifier, the number n of the classifiers and the ordered set C of the classifier can be obtained. After these parameters are determined, the base classifier is run on the test set to output the probability distribution of each classifier to the test samples. We divide the validation set and the test set according to different proportions, which further shows that when the validation set is large enough, the performance of our method on the validation set and the test set can be closer.

![Figure 2. The experimental process.](image)

4.3. Results and Analysis
Table 2 shows the experimental results on the cifar10 dataset. We select the average accuracy of the classifier (that is, the accuracy of each base classifier is added and then averaged) and the existing weighted classification accuracy as a comparison. When dividing the validation set and the test set in the ratio of 1:9, we select different numbers of the best base classifiers to integrate on the validation
set, and determine that the number n of the best base classifiers is 3. Therefore, selecting the first three base classifiers to integrate on the test set, we can see that the accuracy of this method is higher than the average accuracy Ave_ACC is improved by 12.63%, which is 0.65% higher than the existing weighting accuracy. By dividing the validation set and the test set in different proportions, it can be observed from table 2 that the number n of the best base classifiers selected on the validation set is relatively stable, and the number n of the best base classifiers selected on the validation set and the number n determined on the test set.

Table 2. Comparison of CIFAR10 dataset experimental results.

| Algorithm | 1:9  | 2:8  | 3:7  | 4:6  | 5:5  |
|-----------|------|------|------|------|------|
| AVE_acc   | 77.73| 77.69| 77.77| 77.76| 77.57|
| WMV_acc   | 89.71| 89.56| 89.79| 89.85| 89.74|
| Our_method: |      |      |      |      |      |
| n=1       | 82.20| 81.87| 82.15| 81.63| 82.01|
| n=2       | 83.50| 82.99| 82.95| 83.05| 82.93|
| n=3       | 91.00a | 90.36b | 91.25 | 90.29 | 90.37 |
| n=4       | 90.60| 90.27| 90.90| 90.16| 90.37|
| n=5       | 90.80| 89.99| 90.95| 89.84| 90.20|
| n=6       | 90.80| 89.94| 90.65| 89.88| 89.87|

*aThe best accuracy on the validation set when n is different.
*bThe best accuracy on the test set when n is different.

Table 3 shows the experimental results on the cifar100 dataset. The accuracy of this method is better than the average accuracy Ave_ACC is improved by 17.2%, which is 0.91% higher than the existing weighting accuracy. When dividing the test set and the validation set in the ratio of 1:9, the selection number n of the best base classifier is determined as 5 on the validation set and 6 on the test set. Our analysis is affected by the scale of the validation set. With the expansion of the scale of the validation set, that is, through the adjustment of the division proportion, the n determined on the validation set will tend to be consistent with the n determined on the test set.

Table 3. Comparison of CIFAR100 dataset experimental results.

| Algorithm | 1:9  | 2:8  | 3:7  | 4:6  | 5:5  |
|-----------|------|------|------|------|------|
| AVE_acc   | 53.67| 53.66| 53.71| 53.72| 53.81|
| WMV_acc   | 69.96| 69.89| 70.00| 70.03| 70.36|
| Our_method: |      |      |      |      |      |
| n=1       | 65.50| 64.00| 64.75| 64.00| 64.17|
| n=2       | 66.30| 64.56| 65.15| 64.59| 64.60|
| n=3       | 70.10| 69.53| 70.75| 69.41| 70.37|
| n=4       | 71.10| 69.77| 71.35| 69.73| 70.67|
| n=5       | 71.70| 70.07| 71.55| 69.96| 70.80|

5
Our_method: 71.00 70.87 70.78 70.90 71.10 70.95 70.94 71.08
n=6

Table 4 shows the experimental results on the existing dataset. The accuracy of this method is better than the average accuracy Ave_ACC is improved by 22.13%, which is 0.78% higher than the existing weighting accuracy.

Table 4. Comparison of Existing dataset experimental results.

| Algorithm | 1:9 | 2:8 | 3:7 | 4:6 | 5:5 |
|-----------|-----|-----|-----|-----|-----|
| AVE_acc   | 53.72 | 53.77 | 53.81 | 53.94 | 53.98 |
| WMV_acc   | 75.07 | 75.17 | 75.20 | 75.01 | 75.25 |
| n=1       | 71.50 | 73.04 | 73.17 | 72.67 | 72.98 | 73.04 | 72.57 | 73.20 |
| n=2       | 71.33 | 73.31 | 71.71 | 73.47 | 72.67 | 73.26 | 73.08 | 73.10 | 72.68 | 73.55 |
| n=3       | 73.75 | 75.77 | 74.58 | 75.89 | 75.17 | 75.80 | 75.52 | 75.65 | 75.38 | 75.85 |
| n=4       | 73.58 | 75.68 | 74.50 | 75.80 | 74.97 | 75.80 | 75.31 | 75.63 | 75.12 | 75.92 |
| n=5       | 73.75 | 75.82 | 74.79 | 75.97 | 75.11 | 75.94 | 75.46 | 75.79 | 75.22 | 76.18 |
| n=6       | 73.75 | 75.85 | 74.71 | 75.90 | 75.14 | 75.87 | 75.44 | 75.78 | 75.18 | 76.08 |

5. Conclusion and Future Work
In this paper, we proposed a dynamic voting method on the output of models to achieve higher performance. Compared with the existing methods, our one can dynamically select the number of models, which can better fit the dataset. Furthermore, our weight system is based on the probabilities of labels on a sample, which can achieve higher performance than just utilizing the results of models on a sample. In the future work, we will try to apply our method with the combination of other information. For example, the distribution of labels in an environment [25] can be used to further improve the accuracy of the classification.

Acknowledgments
National Natural Science Foundation of China (Grant nos. 61802279, 6180021345, 61702281, and 61702366), Natural Science Foundation of Tianjin (Grant nos. 18JCNJC70300, 19JCTPJC49200, 19PTZWHZ00020, and 19JCYBJC15800), Fundamental Research Funds for the Tianjin Universities (Grant no. 2019KJ019), and the Tianjin Science and Technology Program (Grant no. 19PTZWHZ00020) and in part by the State Key Laboratory of ASIC and System (Grant no. 2021KF014) and Tianjin Educational Commission Scientific Research Program Project (Grant nos. 2020KJ112 and 2018KJ215).

References
[1] Kim M, Han D and Rhee J-K K 2021 Multiview variational deep learning with application to practical indoor localization J. IEEE Internet of Things 8 12375-83
[2] Li J L, Luo G Y, Cheng N, Yuan Q, Wu Z H, Gao S and Liu Z H 2019 An end-to-end load balancer based on deep learning for vehicular network traffic control J. IEEE Internet of Things 6 953-66
[3] Wu Y F, Mendis G J and Wei J 2021 A practical decentralized deep learning paradigm for internet-of-things applications J. IEEE Internet of Things 8 9740-52
[4] Jin G H, Liu F, Wu H, and Song Q Z 2020 Deep learning-based framework for expansion, recognition and classification of underwater acoustic signal J. Experimental and Theoretical Artificial Intelligence 32 205-18

[5] Xin S H, Shi H B, Jide A, Zhu M Y, Ma C and Liao H E 2020 Automatic lesion segmentation and classification of hepatic echinococcosis using a multiscale-feature convolutional neural network J. Medical and Biological Engineering and Computing 58 659-68

[6] López-Gil J M and Garay-Vitoria N 2021 Photogram classification-based emotion recognition J. IEEE Access 9 136974-84

[7] Islam S U, Jan S, Waheed A, Mehmoed G, Zareei M and Alanazi F 2022 Land-cover classification and its impact on Peshawar’s land surface temperature using remote sensing J. Computers, Materials and Continua 70 4123-45

[8] Lécun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition J. Proceedings of the IEEE 86 2278-324

[9] Indolia S, Goswami A K, Mishra S P and Asopa P 2018 Conceptual understanding of convolutional neural network - a deep learning approach J. Procedia Computer Science 132 679 – 88

[10] Hossain M S, Betts J M and Paplinski A P 2021 Dual Focal Loss to address class imbalance in semantic segmentation J. Neurocomputing 462 69-87

[11] Dong X B, Yu Z W, Cao W, Shi Y F and Ma Q L 2020 A survey on ensemble learning J. Frontiers of Computer Science 14 241–58

[12] Kuncheva L I, Whitaker C J, Shipp C A and Duin R P W 2003 Limits on the majority vote accuracy in classifier fusion J. Pattern Analayis and Applications 6 22–31

[13] Kuncheva L I 2014 Combining pattern classifiers: methods and algorithms,Second Edition (New York: Wiley Blackwell)

[14] Kuncheva L I and Rodríguez J J 2014 A weighted voting framework for classifiers ensembles J. Knowledge and information systems 38 259–75

[15] Mu X Y, Lu J F, Watta P and Hassoun M H 2009 Weighted voting-based ensemble classifiers with application to human face recognition and voice recognition Proc. Int. Conf. on Neural Networks (New York: Institute of Electrical and Electronics Engineers) pp 2168–71

[16] Moustafa S, ElNainay M Y, Makky N E and Abougabal M S 2018 Software bug prediction using weighted majority voting techniques J. Alexandria Engineering 57 2763-74

[17] Dogan A and Birant D 2019 A weighted majority voting ensemble approach for classification 4th Int. Conf. on Computer Science and Engineering (New York: Institute of Electrical and Electronics Engineers) pp 366-71

[18] Tasci E, Uluturk C and Ugur A 2021 A voting-based ensemble deep learning method focusing on image augmentation and preprocessing variations for tuberculosis detection J Neural Computing and Applications 33 15541–55

[19] Lee Y W, Hwang J-W, Lee S, Bae Y and Park J 2019 An energy and GPU-computation efficient backbone network for real-time object detection 32nd IEEE/CVF Conf. on Computer Vision and Pattern Recognition (New York: Institute of Electrical and Electronics Engineers) pp 752-60

[20] Simonyan K and Zisserman A 2015 Very deep convolutional networks for large-scale image recognition arXiv:1409.1556v6

[21] Gao S H, Cheng M M, Zhao K, Zhang X Y, Yang M H and Torr P 2021 Res2Net: A New Multi-Scale Backbone Architecture J. IEEE Transactions on Pattern Analysis and Machine Intelligence 43 652-62

[22] Ding X H, Zhang X Y, Ma N N, Han J G, Ding G G and Sun J 2021 RepVGG: making vgg-style convnets great again arXiv:2101.03697v3

[23] He K M, Zhang X Y, Ren S Q and Sun J 2016 A deep residual learning for image recognition Proc. of 2016 IEEE Conf. on Computer Vision and Pattern Recognition (New York: Institute of Electrical and Electronics Engineers) pp 770-8
[24] Huang G, Liu Z, Van Der Maaten L and Weinberger K Q 2017 Densely connected convolutional networks 30th IEEE Conf. on Computer Vision and Pattern Recognition (New York: Institute of Electrical and Electronics Engineers) pp 2261-69

[25] Jin G H, Hu Y X, Jiao Y M, Wen J F and Song Q Z 2021 Improving the performance of deep learning model-based classification by the analysis of local probability J. Complexity 2021 11 pages